2006

Development of an Automated Template Selection and Alignment Tool for Protein Structure Homology Modeling

David R. Riley

Development of an Automated Template Selection and Alignment Tool for Protein Structure Homology Modeling

Approved: Jiye Shi
Thesis Advisor

Gary Skuse
Director of Bioinformatics or
Head, Department of Biological Sciences

Submitted in partial fulfillment of the requirements for the Master of Science degree in Bioinformatics at the Rochester Institute of Technology.

David R. Riley
May 2006

Thesis/Dissertation Author Permission Statement

Title of thesis or dissertation: Development of an Automated Template Selection and Alignment Tool for Protein Structure Homology Modeling

Name of author: David R. Riley
Degree: Master's of Science
Program : Bioinformatics
College: Science

I understand that I must submit a print copy of my thesis or dissertation to the RIT Archives, per current RIT guidelines for the completion of my degree. I hereby grant to the Rochester Institute of Technology and its agents the non-exclusive license to archive and make accessible my thesis or dissertation in whole or in part in all forms of media in perpetuity. I retain all other ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of
this thesis or dissertation.

Print Reproduction Permission Denied:

I, David R. Riley, hereby **deny permission** to the RIT Library of the Rochester Institute of Technology to reproduce my print thesis or dissertation in whole or in part until after 31 May 2007. Thereafter permission is granted.

Signature of Author: David Riley

Date: 12 May 2006

Inclusion in the RIT Digital Media Library Electronic Thesis & Dissertation (ETD) Archive

I, David R. Riley, additionally grant to the Rochester Institute of Technology Digital Media Library (RIT DML) the non-exclusive license to archive and provide electronic access to my thesis or dissertation in whole or in part in all forms of media in perpetuity but not until after 31 May 2007. I understand that my work, in addition to its bibliographic record and abstract, will be available to the world-wide community of scholars and researchers through the RIT DML. I retain all other ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation. I am aware that the Rochester Institute of Technology does not require registration of copyright for ETDs.

I hereby certify that, if appropriate, I have obtained and attached written permission statements from the owners of each third party copyrighted matter to be included in my thesis or dissertation. I certify that the version I submitted is the same as that approved by my committee.

Signature of Author: David Riley

Date: 12 May 2006

Thesis Advisory Committee Members:

Dr. Jiye Shi

Principal Scientist

UCB Pharmaceuticals

Dr. Gary Skuse

Director of Bioinformatics

RIT Department of Biological Sciences

Dr. David Lawlor

Associate Professor

RIT Department of Biological Sciences

Dr. Paul Craig

Professor of Biochemistry

RIT Department of Chemistry

Abstract:

DNA and protein sequence data sets have exploded in size over the past several years and have stimulated the development of new methods for closing the sequence/structure gap. Bench laboratory methods, while being the most accurate, are still far too time consuming and limited to breech this gap. Thus computational methods have become the standard. The work described here details the development of a system to automatically select a structural template, align the template and query sequence and model the query. The system was implemented fully for antibodies but maintains a modular design that can accommodate a wider variety of molecule types. Extensive testing was performed using antibodies and the results of this analysis reveal the potential limitations of homology modeling and point toward ways to improve on the template selection method.

List of Figures:

List of Tables:

List of Appendices:

Content:

Introduction:

The explosion of DNA and Protein sequence data has stimulated the development of new more sophisticated methods for determining 3-dimensional protein structure from 1-dimensional sequence. "Protein structure determination is one of those areas of biology where both experimental and theoretical approaches have made a lot of progress in recent years and are complementing each other" (Peitsch 2001). Theoretical approaches typically employ computer programs which attempt to find a structure either using physical constraints or homologs with known structure. The powerhouse experimental or "wet lab" methods in this effort are X-ray crystallography and NMR spectroscopy.

X-ray crystallography is the foremost technique in protein structure determination. This method has contributed 25,774 of the 29,757 protein structures recorded in the Protein Data Bank as of September 13th 2005 (Berman 2000). This technique requires a highly purified sample of the protein which is crystallized using one of any number of crystallization methods. The crystal locks the protein in a regular lattice structure. The crystal is then illuminated with a beam of x-rays. This beam is scattered, mostly by the electrons around each atom and to a small extent by the nucleus of each atom. The scattered x-rays strike a detector and produce a diffraction pattern. The pattern produced by the scattered x-rays is read and analyzed. This pattern is mathematically extrapolated into three dimensions and then refined several times so as to avoid violating known physical constraints. The major advantage of this technique is its resolution. X-ray structures can often be refined to resolutions below 2.0 Å. The computational refinement process has improved markedly since the techniques inception; however the crystallization step, arguably the most important and difficult, has not seen the same rapid advancement. The first International Conference on Crystallization of Biological Molecules was in 1984 and "after 20 years, the bottleneck step in obtaining protein structures by crystallographic methods is still at the

crystallization level" (Pusey *et al.* 2005). As a result, the number of resolved structures is dramatically smaller than the number of sequences; this fact is illustrated in figure 1. In addition, many samples (approximately 90% of all potential crystal structures) have reached the point of high purification but have been discarded because they do not crystallize well. This is a critical limitation in this technique and one reason why structure databases like the PDB have only a fraction of the entries found in sequence databases. The Protein Information Resource (PIR) Non-Redundant Reference Protein Database (PIR-NREF) contains all the entries from the PIR, SwissProt, TrEMBL, RefSeq, GenPept and the PDB and contains 2,504,019 entries compared to the PDB's 29,757 (Wu *et al.* 2002).

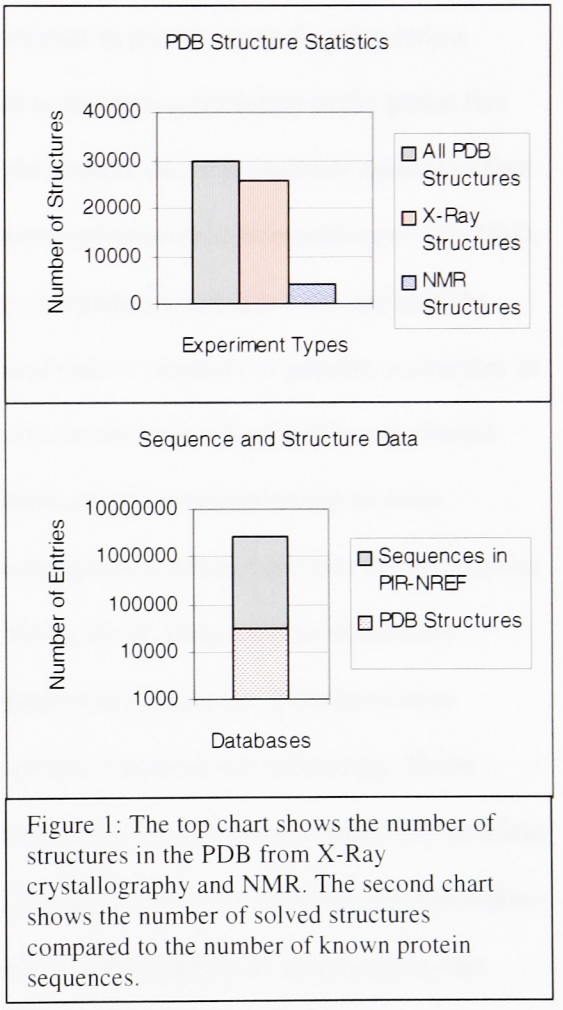


Figure 1: The top chart shows the number of structures in the PDB from X-Ray crystallography and NMR. The second chart shows the number of solved structures compared to the number of known protein sequences.

An alternative wet lab method is NMR Spectroscopy. In this method a highly purified protein sample is placed in a magnetic field and the reaction of nuclei to this field is measured. NMR has some advantages over X-ray crystallography in that it does not require a protein to be crystallized. "Most (75%) of the NMR structures in the Protein Data Bank (PDB) do not have corresponding crystal structures" (Montelione *et al.* 2000). As was mentioned previously, the most significant drawback to X-ray crystallography is the crystallization process. NMR avoids this complication at the expense of resolution. "The highest quality NMR structures have accuracies comparable to 2.0-2.5 Å X-ray crystal structures" (Montelione et al 2000). As a result, NMR structures are useful in many applications where X-ray structures are not available. However,

NMR has a molecule size restriction, the equipment is expensive and as a result this technology has yet to make inroads into the sequence/structure knowledge gap. The Protein Data Bank has 3,983 (of 29,757 total structures) NMR structures as of September 13th 2005 (Wu *et al.* 2002).

Because of the costs associated with X-ray crystallography and NMR, many researchers have turned to *in silico* methods for structure determination. There are two major techniques in computational structure prediction: *ab initio* and comparative (homology) modeling.

Ab initio methods are based solely on physics and attempt to predict structure using only a sequence. The assumption made here is that the native structure corresponds to the global free energy minimum accessible during the lifespan of the protein. *Ab initio* methods attempt to find this minimum energy conformation by exploring numerous possible conformations (Fiser 2004). The ideal would be to iteratively try every possible conformation and select the one with the lowest free energy. This method is theoretically sound since it attempts to generate a structure in much the same way a protein is actually folded *in vivo*; however it is hindered by our limited understanding of exactly how proteins fold. As a result, models produced using *ab initio* techniques are typically of low resolution. Compounding this problem is the fact that the physics involved in protein folding is massively complex. With a tremendous number of possible combinations, this problem quickly becomes computationally strenuous. Proteins of even moderate length cause this technique to overflow current computational technology. While projects such as Stanford's Folding@Home and IBM's Blue Gene attempt to attack this problem with massive computational power, the inaccuracy and uncertainty in the folding process makes this technique seem like merely a pipe dream. Stanford's Folding@Home site speaks to the challenges:

...it takes about a day to simulate a nanosecond (1/1,000,000,000 of a second). Unfortunately, proteins fold on the tens of microseconds' timescale (10,000 nanoseconds). Thus, it would take

10,000 CPU days to simulate folding – i.e. it would take 30 CPU years! That's a long time to wait for one result (Pande 2000).

The extreme computational limitations of *ab initio* methods drive most researchers to choose comparative modeling. Unlike *ab initio* methods which are based solely on physics, comparative or homology modeling techniques are based on evolutionary relatedness. In this method a template structure is selected that is closely related to the structurally unknown query sequence. The template and query sequences are aligned and the query sequence is threaded around the template structure. Once the query has been threaded most programs refine the model by enforcing physical constraints, much like those used in *ab initio* approaches. Comparative modeling has become the most popular and accurate means of predicting structure *in silico*. Models generated using homology modeling can have resolutions comparable to low-resolution x-ray crystallography or medium-resolution NMR (Fiser 2004). If a close homolog is chosen, RMSD values of 0.5 and 0.8 can be attained for templates with 2 Å and 3 Å resolutions respectively.

One of the keys to the success of a comparative modeling experiment is the informative selection of a structural template. While many scientists rely on database searches such as BLAST, PSI-BLAST, and FASTA these searches examine only sequence similarity. Although one can argue that sequence similarity is very important to the overall quality of a resulting model, the structural/functional implications of each mutation are not taken into account with these methods. More specifically all positions in the sequence are considered equal. Evolutionarily speaking, this assumption fails in almost all protein families. This concept will be illustrated in antibody, where certain regions have particularly high mutation rates while other regions remain well conserved. It is for this reason that the importance of template selection not be underestimated.

Once a suitable template has been selected, or in some cases during the selection process, the unknown sequence and template must be aligned. This alignment step can prove difficult if considerable care is not taken to ensure that conserved regions are properly aligned. In some cases regions of high variability can incorrectly displace a portion of the unknown sequence. This dramatically affects the quality of the model because sequentially conserved regions are also structurally well conserved. Consequently regions of high variability can differ in both length and amino acid composition without dramatic changes to the overall structure. Likewise alignment of conserved regions should be maintained if at all possible.

Homology modeling is based on the structural conservations of the framework regions between the members of a protein family. The 3D structures are more conserved in evolution than sequence, and hence even the best sequence alignment methods frequently fail to correctly identify the framework regions that possess the desired level of sequence similarity (Prasad *et al.* 2003).

In addition to the reliance on a quality structure alignment, homology modeling software faces another challenge. Because so many fewer structures than sequences have been realized (refer to figure 1) not all proteins will have evolutionary homologs with solved structures. Additionally, even fewer structures have been solved at high resolution. As was mentioned previously the best resolution structures come from X-ray crystallography studies. Because this technique is limited to proteins that will crystallize well, the data set of high resolution protein structures is quite biased and limited.

That being said, when considering all *in silico* methods, comparative modeling is the only one that can reliably generate an accurate model of a protein from its amino acid sequence (Schwede 2003). Homology modeling provides the greatest potential for expedited growth of derived structure databases. This potential is amplified in light of the Protein Structure Initiative suggested in 1999. This initiative called for an effort in line with the Human Genome project in which X-ray crystallographic structures and NMR solutions are produced at high throughput for

every globular protein family. The result of such an effort would be a library of structures which could be used as templates for homology modeling of nearly any globular protein sequence (Burley *et al.* 1999). Until such complete libraries are realized, homology modeling efforts continue utilizing protein families with experimentally determined structures.

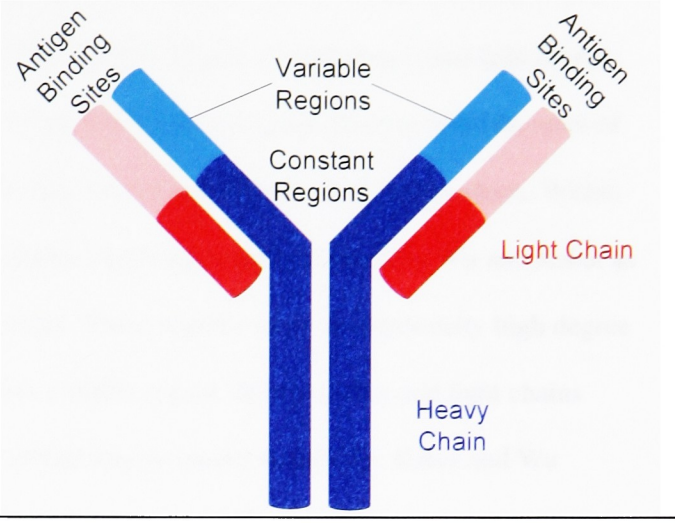


Figure 2: The basic structure of an entire antibody is shown here. The heavy chain is shown in blue and the light chain is shown in red. Constant regions are darker in color while variable regions are light in color. The antigen binding sites are located at the ends of the variable regions.

Antibody proteins have been of considerable interest in the pharmaceutical industry for many years. The specificity of binding and the ability to manipulate this binding have made antibody a popular tool for drug discovery. Antibody proteins are made up of 4 chains, two heavy and

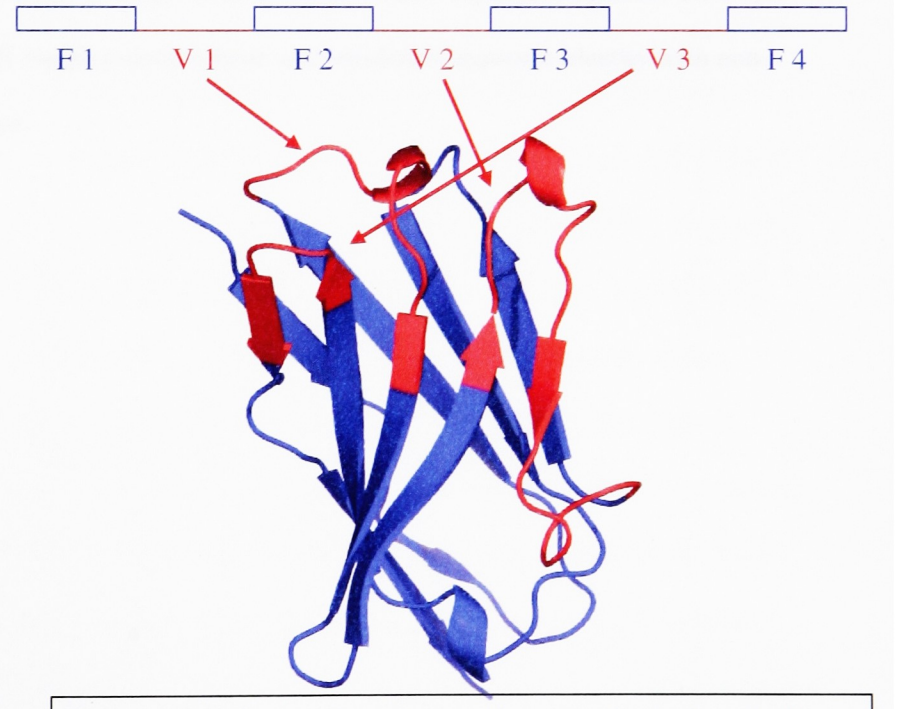


Figure 3: This diagram shows the variable region of a single antibody chain. There are 4 framework regions and 3 hypervariable regions structured as follows (V- Hypervariable region, F- Framework region). Note that all the variable regions are located on the top of the structure. This is the site of antigen binding. The boxes and lines above the structure show the order of the frameworks and hypervariable regions in sequence. (Visualization created using PyMol (DeLano 2002))

two light. They combine to make a Y shape as shown in figure 2. Additionally, antibody chains have two regions: a variable and a constant. The variable regions of both heavy and light chains are at the tips of the Y. The constant region of the heavy chain extends down to form the stem of the Y while the light chain constant region simply extends to the kink in the heavy chain. Within the variable region on both heavy and light chains there exist regions defined by Wu and Kabat as "Complementarity Determining Regions" (1970). These regions show an abnormally high degree of variability when compared to the rest of the variable region. In both heavy and light chains there appeared to be three regions which exhibited this excessive variability. Kabat and Wu suggested a numbering system that continues to be used today. This numbering system assigns a number to each residue in an antibody variable region and defines the boundaries of the Complementarity Determining Regions (CDRs or Hypervariable Regions). This system is extremely useful when comparing antibody variable regions since sequence alignment tools can sometimes be fooled by CDR length polymorphism and incidental sequence identity with non-CDR or Framework Sequence.

Materials and Methods

Data Sets:

The first step in this project was the collection of data sets. There were three major data sets compiled for the final configuration of the system, two containing sequence and one structure. Because the success of the system is based primarily on the quality of these underlying data sets, great care was taken in ensuring their quality. In several cases structures and/or sequences with misannotation, incomplete data and/or particular abnormality were removed from the data set to avoid tainting the results.

One sequence data set was compiled according to the subtype categories developed at the International ImMunoGeneTics Information System or IMGT (Lefranc *et al.* 2005). Subtyping and chain identification are important to keeping track of what type of antibody chain the system is working with. All of the Kappa light chain and heavy chain subtypes were collected from IMGT and these sequence sets were each aligned using ClustalW (Thompson *et al.* 1994). These alignments were of whole variable regions and included insertions and CDR residues. Containing 8,250 sequences in all, the alignments were then converted into Hidden Markov Models using HMMER's hmmbuild program (Eddy 2005). These HMMs were then combined creating a system which would allow HMMER's hmmpfam search tool to take any antibody chain and predict its chaintype and subtype. The speed at which hmmpfam returns these subtypes awards an extraordinary amount of flexibility. Chain types and subtypes can be assigned quickly for both query and database antibody chains, adding to the intelligence of the system and the ability to minimize the amount of information the user must input for the system to work. A PERL wrapper class was also written to easily handle running the hmmpfam program and returning the results in a standardized format. This wrapper will be discussed more fully in the implementation section,

but its major purpose was to provide a layer of abstraction that resulted in faster development time and reduced maintenance.

The second sequence data set was collected for use in ClustalW Profile alignments. The purpose of these alignments was to provide a way to quickly and effectively assign Kabat numbers to antibody sequences. Creation of this data set began with aligning all sequences from antibody heavy and Kappa light chains using ClustalW. In this set, subtypes for each chain type are all lumped together and aligned. This produced ClustalW alignment profiles. These profiles were manually edited so that they included only the whole number Kabat residues (i.e. 23, 24, 25 not 23A or 23B). In addition, CDR residues were removed and replaced with gap characters ('-'). This created a gapless alignment of all the antibody frameworks. The entire processing step is depicted in figure 4 with one sequence.

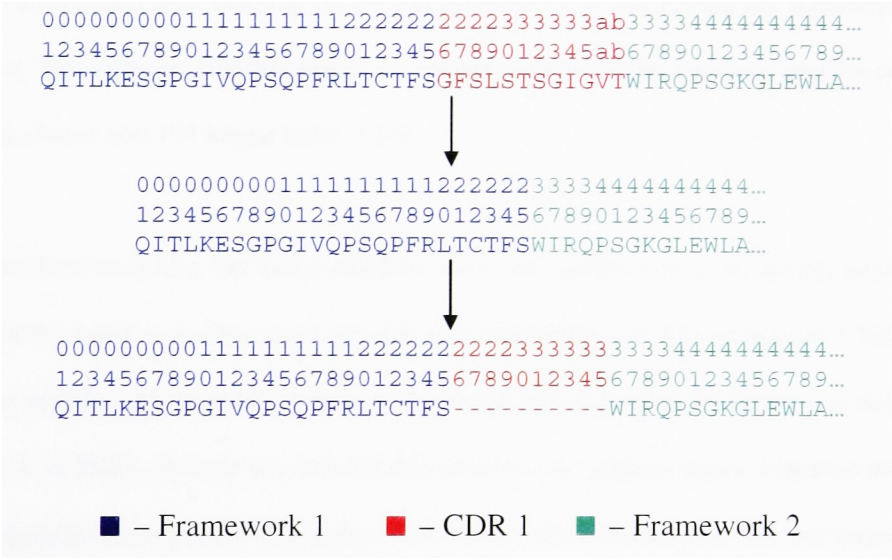


Figure 4: The protocol for processing the ClustalW profile alignments is shown here. The CDR region is removed and replaced with the exact number of '-' characters. An example of the resulting profile is shown in Appendix 1.

Any sequences with framework deletions were removed from this profile so as to not introduce gaps. The resulting profile contained only framework sequence and the proper number of gaps to constitute a gapless CDR. The purpose of this procedure was to remove CDR sequence that could

PID	Heavy Chains	Light Chains
Unfiltered	217	811
100	187	659
95	81	379
90	39	101
80	6	18

Table 1: These are the numbers of sequences that resulted from filtering the heavy and light chain antibody profiles. 100% in heavy chain and 90% in light chain were used in the final profiles.

potentially mislead the alignment protocol by introducing false residue matches. Since CDR sequence is particularly variable, the chance exists for CDR sequence to match well enough with framework sequence to shift the alignment and cause improper residue numbering. This risk is particularly high when there are insertions and/or deletions in the query sequence. Additionally, because the goal of this system is the modeling of framework residues, the sequence within a CDR should be considered only in the structural context. That is, the identity of residues at each position within a CDR is not critical to the performance of the system. An example of the resulting profile alignment is shown in Appendix 1. The actual profile was filtered using 100% identity in heavy chains and 90% identity in light chains. This filtering level was determined after filtering on several different PIDs and testing the accuracy of the numbering. The different filtering levels are shown in table 1. The resulting profiles contained 187 heavy chains and 101 kappa light chains.

The protocol for assigning the Kabat numbers using this profile begins by simply passing the profile and the query sequence into ClustalW and requesting a profile alignment. ClustalW then returns the profile with the query sequence appended and aligned at the bottom, as depicted in Appendix 1. A PERL module was written that retrieves the aligned query sequence and the profile sequence directly above the query. The module then iterates over the two sequences and assigns numbers to the residues in the query based on the knowledge that there are no insertions or deletions in the profile sequence. The CDR region is numbered by simply assigning consecutive numbers until there is no more query sequence CDR. If there is an insertion in the query sequence CDR then letters are added to the last number and assigned to these extra residues

(i.e. 35a, 35b etc.). In this way, the identity of CDR residues is ignored and their position relative to the flanking frameworks is maintained.

The third data set key to the system's functionality is the high resolution structure library. Over the course of the project this library underwent several reincarnations each trying to avoid biases and to eliminate abnormal structures and structures containing uncommon antigens. The goal of these changes was to create a set of structures to represent all of the currently solved structures, while eliminating redundancy and any unnatural products of research efforts. To start, structures had to be solved at resolutions of 2.5 Å or better. This limited the set to X-ray crystal structures. The structures were retrieved by searching the PDB for antibody structures. Once the structures were identified the PDB files were retrieved and filtered to avoid redundancy.

The filtering process was done on both heavy and light chains independently such that both sets would be individually nonredundant as well as being nonredundant when combined into chain pairs. When filtered at 90% PID the result was 176 heavy chains and 92 light chains. When combined, the resulting chain pair set contained 81 chain pairs. This indicates that there was some chain sharing between molecules: possibly the result of experimental mutation in one chain and not the other. While this technique does eliminate 94 unique heavy chains and 10 unique light chains, it was important that the system be able to model chain pairs as well as single chains. Therefore it was important to have a consistent data set for both single chain and chain pair modeling. Having a different data set for each test situation would be possible; however it would eliminate the possibility of comparing the results from single chain and chain pair tests. For the purposes of testing and determination of the technique's viability, the 81 heavy/light chain pair structures were used.

Once the structure data were decided upon, they was processed. This step began with replacing the residue numbers in the PDB files with the proper Kabat numbers. This was a way of streamlining the system by eliminating the need to renumber these sequences every time they are used. In addition to Kabat numbering, the constant regions and antigen atoms were removed from the files. This step both eliminated the possibility of interference between the antigen and the model during model generation and shrunk the file size. Any step that helped to reduce memory usage was helpful at this stage. Next the chain type and subtype were placed in REMARK lines directly before the ATOM coordinates in the PDB files. This annotation proved useful later as it ensured that the chain about to be read in was the type that was expected. While the heavy chain typically comes first in these files, one can never assume that it will always be the case.

Template Selection:

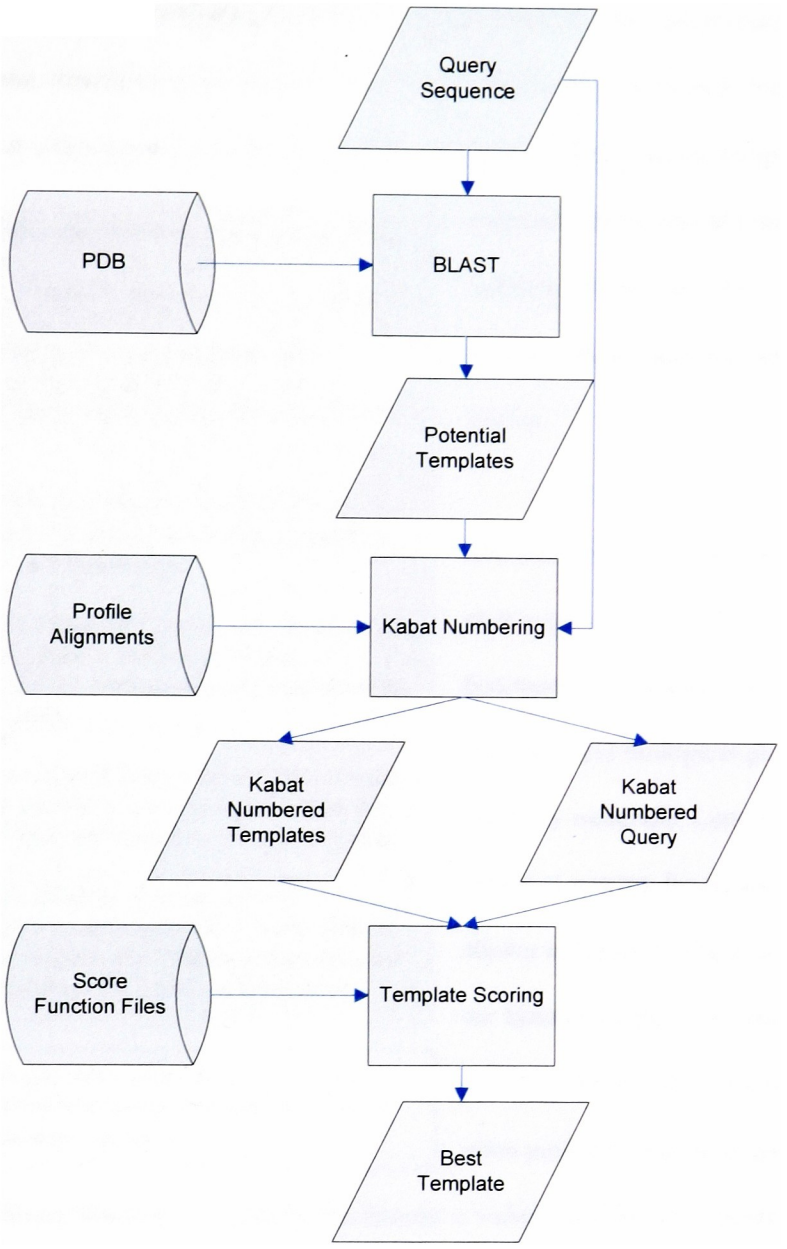


Figure 5: This is an overview of what the system does to select a template. The input to the program is a query sequence of either 1 or more chains which is BLASTed against the PDB. The resulting list of structures is filtered such that only structures in the high resolution set proceed to the next step. The templates and query sequences are numbered according to the Kabat rules using the ClustalW profile alignment. The numbered templates are then scored using the scoring functions specified by the user. The result is a template with the highest score which will be used for modeling.

A graphical depiction of the template selection method is shown in figure 5. This procedure takes as input the query sequence which is BLASTed against the PDB. The search results are filtered to include only those structures in the high-resolution set. This step is performed first to facilitate the use of BLAST as a benchmark and also to quickly generate a list of possible templates. The query sequence and potential templates are Kabat numbered and the comparison between these templates and the query begins.

Score Functions – All give 0 for matches in CDR

A – BLAST: e-values are used to choose a template

B – Identity/Homology: 1 for a match 0 for a mismatch unless a scoring matrix is used in which case the value from the substitution matrix is used

C – CDR length bonus: Gives a bonus if CDR lengths match (or if the lengths are greater than or equal to a certain length)

D – Heavy/Light chain interaction: The distance from the opposite chain is used to generate a score. If a matrix is used the distance score is multiplied by the matrix value.

E – Proximity to CDR: Distance from CDRs is used to generate a score. If a matrix is used then the distance score is multiplied by the matrix value.

F – Solvent Accessibility: Percent solvent accessibility is used to generate a score. Buried residues are emphasized. Matrix values are used for buried residues if a matrix is being used.

Figure 6: The single letter codes for each scoring function are shown here along with a description of each scoring function's purpose.

The templates are scored according to the user's specifications. The different scoring functions are designed to exploit different chemical and biological properties of the antibody molecule. Letters identify the different scoring functions and the key is shown in figure 6. These scoring functions are based on a position specific scheme in which Kabat numbers are used to identify what positions will be emphasized and which will not. Score function A is simply the inverse e-value from BLAST. Score function B assigns a 1 for a match and 0 for a mismatch in frameworks. 0 is assigned to residues in CDRs regardless of whether they match or not. B can also be used with a substitution matrix. When a matrix is used, the value in the matrix is assigned to all framework residues. 0 is still assigned to CDR residues regardless of their identity. Score function B is used in conjunction with all the remaining scoring functions (C-F). Because these scoring functions do not take into account overall sequence similarity, they require a base score on which to build. In this way, these scoring

functions are simply add-ons to the basic homology score function B, and provide increased sensitivity based on the different chemical and biological properties they exploit.

Score function C awards a bonus (by default 10) for each CDR in the template with the same length as the corresponding CDR in the query. This score function was implemented because it appeared as though equal length CDRs could improve a model's overall quality. While this seemed to be the case it became clear that length equality was not sensitive enough to exploit the phenomenon. As a result the technique was adjusted to give the bonus if the length of the template CDR was equal to or greater than the length of the query CDR. While these values were modified several times for training, the default values of 10 for a perfect match and 75% of that or 7.5 for a template with CDR length greater than that of the query. These bonuses were awarded for each CDR region that had these favorable lengths making the maximum bonus 30 and the minimum bonus 0.

Score function D is based on the premise that residues which participate in chain-chain interaction could be important in the overall model quality. In this implementation the distance from each framework residue in one chain to the nearest framework residue of the other chain was measured. This measurement was performed using a custom script which calculated the distance using the x, y and z coordinates from each atom of each residue. The closest two atoms between the target residue and the opposite chain were used. Like score functions C and B, the parameters associated with this scoring function were adjusted several times to find the best weight. In particular this scoring function could be binned so that residues with distances less than 2 Å could receive a score of 3, residues less than 4 could receive a score of 2 and residues with distances less than 6 could receive a score of 1. These distances are based on accepted hydrogen bond lengths and therefore could provide some increased sensitivity. However, this

technique was discarded in the end and simply awarding a score of 1 for identical residues within 6 Å of the opposite chain was used as the default value.

Score function E is based on the 3-dimensional proximity of individual residues to the nearest CDR residue. For this score function a script was written to measure the distance between all atoms in all framework residues and the nearest atom in a CDR residue for all the high resolution structures. The distance between the closest two atoms was taken for each framework position. The average and standard deviation were calculated for each position and then manually examined. Like score function D, a binning system could be used based on hydrogen bond lengths. This system was tested during the training phase, but was replaced with simply assigning a default score for all residues within 6 Å of the nearest CDR residue. This was based on the theory that the loop region structure will be adjusted by the identity of these nearby residues and, in turn, the loops could affect the overall framework structure.

Score function F was developed to take into account the solvent accessibility of each residue position. In this function the solvent accessibility of each framework residue in all of the high resolution set was measured using PSA (Sali *et al.* unpublished). The average and standard deviation of the percent solvent accessibility at each position were calculated and positions having an average of 7% accessibility or less, a commonly accepted threshold, were considered buried. In addition to the average, positions must have had favorable standard deviations which indicated a consistently buried state. It was not worth considering residues with variances that indicated inconsistent orientations as they could not be considered buried for all structures. Those residues that met these criteria were awarded a default score of 1 while all other residues received a score of 0.

This system is easily expandable and the current scoring functions are simple to edit. The score

information is simply a text file with the score values separated with tabs. An example of a score function file is shown in Appendix 2. The file corresponding to the specified score function is read into the system and scores are applied accordingly. Because more than one scoring function may be used, a weight can be applied to help normalize these scores. Additionally, weights can be applied for each chain type to further customize the way templates are scored. Once all the specified scoring functions have been assessed, the template with the highest score is selected as the best. In the case of ties, one of the best templates is used and the remaining ties are reported as such. One convenient feature of this scoring system is the ability to see which scoring functions contributed what values to the overall score of each template. The program prints several files containing the position specific scores for all templates for each scoring function specified. This assisted in troubleshooting the program and gives the user the ability to examine the affect of each scoring function.

Alignment/Modeling/Superimposition:

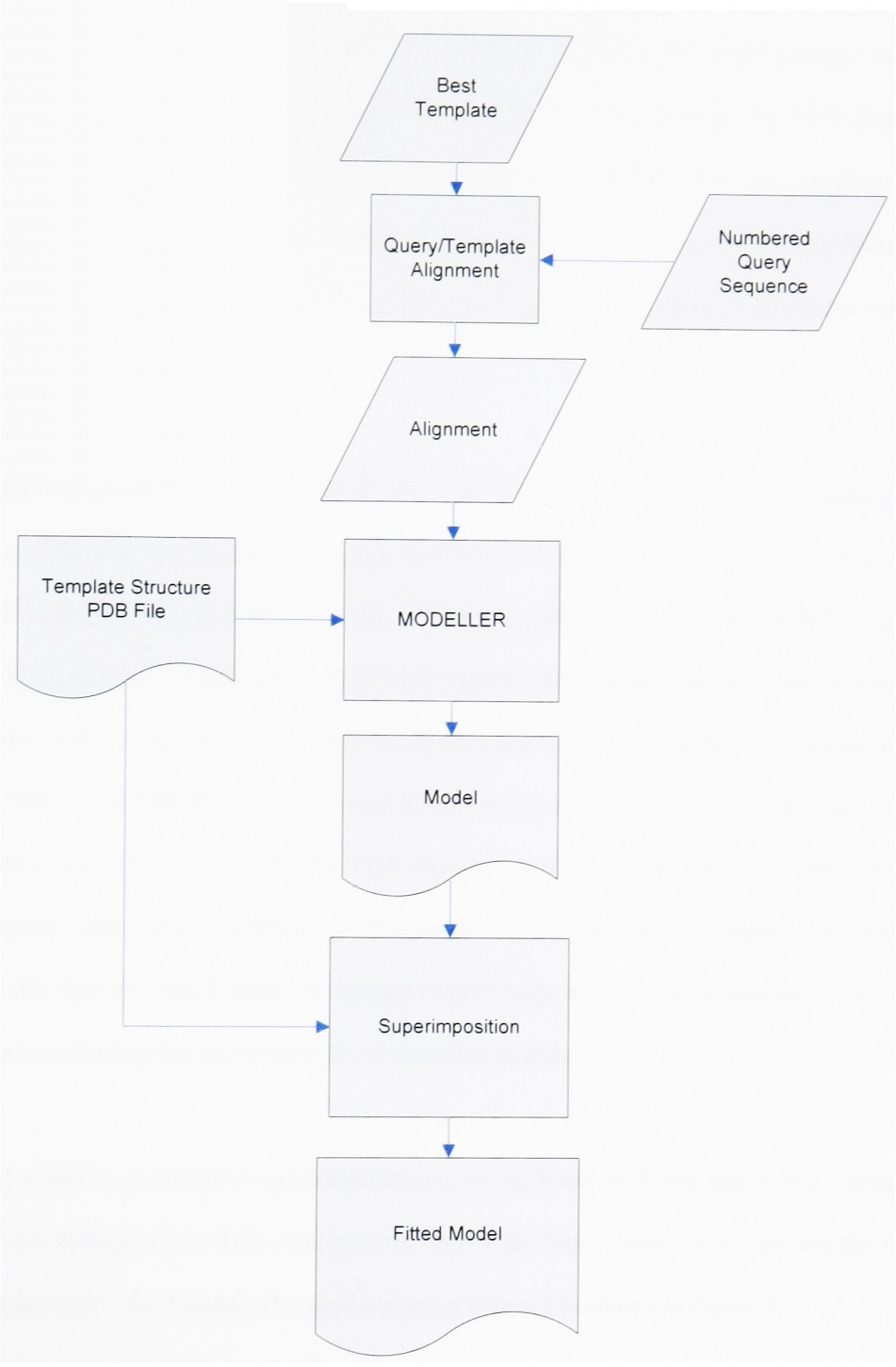


Figure 7: This is an overview of what the system does once it has selected a template. This begins with aligning the template and query sequence. This step is simple since both have been Kabat numbered. This alignment is then passed into MODELLER along with the template structure. The resulting model is the superimposed with the template structure to make them easier to compare.

Once a template is selected the next step is alignment of the query and template sequences. In earlier versions of the program a ClustalW alignment was used for this step. However, as mentioned earlier, this alignment method can be fooled by incorrectly aligning CDR sequence with framework. Additionally this step is unnecessary since both the query and template sequences are Kabat numbered. As a result, a routine was written that takes the two Kabat numbered sequences and generates a PIR formatted alignment file based on the Kabat residue numbers.

At this point the system has done the difficult work of choosing a template and preparing an alignment. The next step in the process is executing the modeling program. In this version of the program Accelrys' MODELLER was used. An attempt was also made to utilize NEST, part of the JACKAL package (Xiang 2002). While this attempt showed the ease at which an alternative modeling method could be incorporated, the models generated from NEST were decidedly sub-par. For this reason MODELLER was used for all testing and exploration. For the sake of this project the refinement level in MODELLER was set low. Because we were only interested in modeling the alpha carbon backbone, a refinement level beyond this was unnecessary and would increase the runtime considerably. In a project where large data sets were planned to be processed, increasing this refinement level would be impractical.

Once the model is created it is superimposed over an antibody molecule that is in a standard 3-dimensional configuration. This configuration shows the heavy chain on the left and the light chain on the right. An example of superimposed structures is shown in figure 8.

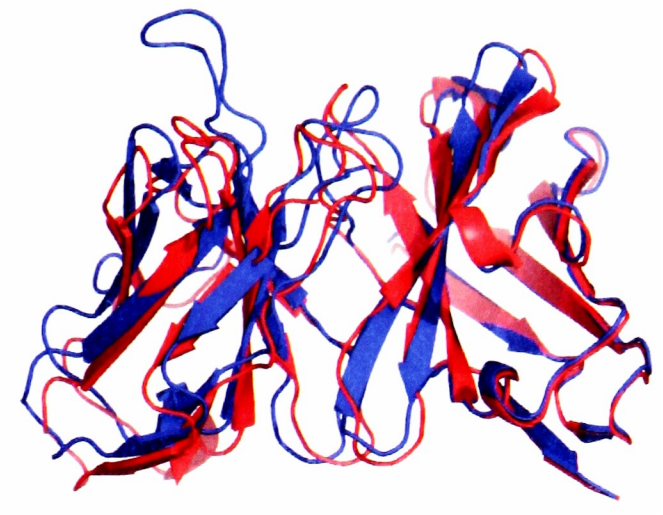


Figure 8: This is an example of two structures, one in blue and one in red, that have been superimposed using the ProFit superimposition program and the runprofit.pl script.

The superimposition method used for this project was ProFit from Andrew Martin's Lab at University College London (Martin 2005). ProFit takes as input two structures and specifications as to what atoms should be used for the superimposition. In this application only alpha carbons from amino acids in the antibody framework are used to perform the superimposition. This ensures that structural variation in the CDRs will not effect the position of the superimposed structure. While this step does not affect the quality of the model it does place the structure in a standard orientation for viewing in molecular visualization software.

Program Design

In order to achieve the level of functionality described above, the system needed to be designed to eliminate redundancies and allow for easy adjustment and expansion. To accomplish this, the system was designed as separate parts that were then combined in a pipeline style. One key element of the design was the implementation of a pair of PERL objects to store the complex data associated with chains and molecules. For the sake of this implementation, a 'molecule' is a collection of chains that are contained within a single structure (i.e. an antibody heavy and light chain pair). A 'chain' is an individual continuous sequence of amino acids (i.e. a heavy or light chain). The diagram in figure 9 shows the relationship between these two objects. The molecule object contains all of the specific information about a given type (i.e. antibody or perhaps kinase, etc.). The chain object is given all of the chain specific information from the molecule object.

This means that there is only chain specific information in one place, the molecule object. As a result, a chain must be associated with a molecule from which it will be given important information like the location of high resolution structure files for the individual chain, CDR boundary locations for the chain type and the order in which this chain appears in the PDB file.

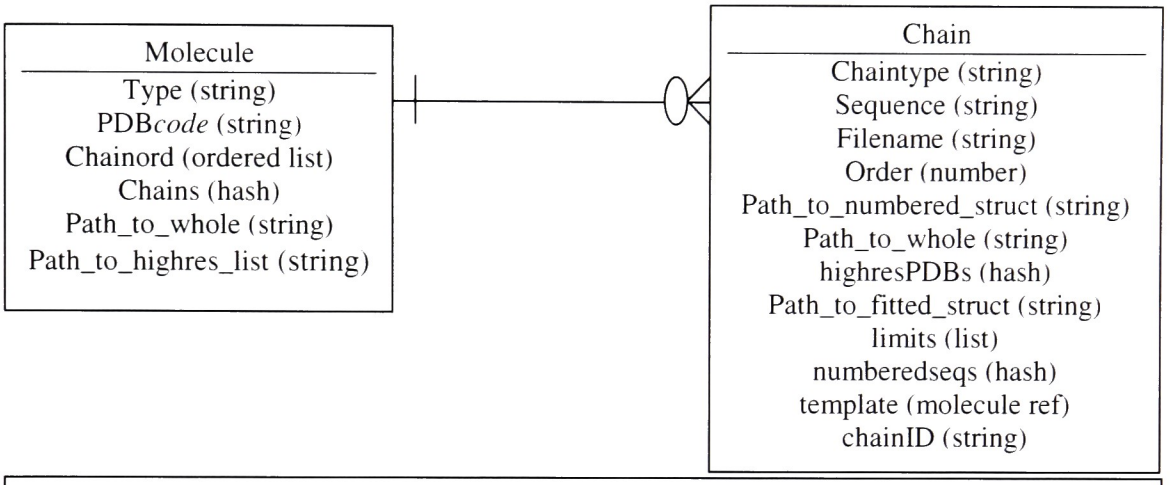


Figure 9: This is a diagram of the classes Molecule and Chain. The molecule object contains information specific to a given molecule type (aka. antibody or kinase). The Molecule object also contains the chain specific data which is used to create chain objects which will contain residue numbered sequence, hypervariable region boundaries, etc.

One of the most important and useful elements of the Chain object is the numberedseq hash variable. This variable is a PERL hash which is an associative data structure. In this particular instance the hash is used to store the Kabat numbers as keys which refer to the associated 1-letter residue identifier. A sample of what the numberedseq hash looks like conceptually is shown in figure 10. Functionally this provides a simple way of storing the Kabat numbered sequence from any antibody chain. This data structure is simple to manipulate and compare, which eases the scoring and alignment steps of the modeling procedure. Additionally, the ability to pass references of this data structure reduces memory usage and improves performance.

Numberedseq hash:

```
33 : G
34 : I
35 : G
35a: V
35b: T
36 : W
```

Figure 10: This is a conceptual representation of what the numberedseq hash looks like. The Kabat numbers on the left are keys to the 1 letter residue codes on the right.

Once this object model was established, the next step was to integrate the objects into the routine. This step was accomplished both with adjustment of the input sequence handling and with a script known as pdb2mol.pl. The input query chain sequence(s) is/are immediately made into a molecule object with associated chain objects. This adjustment was made in the control script that directs the program flow. Additionally, the pdb2mol.pl PERL code was written to take any PDB structure and create a Molecule object with its associated Chain objects. This procedure greatly simplified the data collection step. Once the potential templates were determined their PDB files were used to create associated Molecule objects and these objects were used for scoring and alignment. The ability to pass Molecule object references around greatly reduced the memory and disk I/O lag by eliminating the need to constantly read PDB files. In addition, the pdb2mol.pl script added another layer of abstraction that makes expansion to other molecule types much simpler.

Two additional PERL scripts were integral to the system: kabatPDB.pl and runprofit.pl. Each of these scripts is a wrapper for another program. The kabatPDB.pl script takes as input an antibody PDB file and outputs the same PDB file with chain identification REMARKS added before each chain and Kabat numbers substituted in the residue number column. This script masks the ClustalW profile alignment step, the hmmpfam subtyping routine and makes the Kabat numbering of any antibody structure a routine task. This script was crucial when it came time to annotate the high resolution data sets.

The runprofit.pl script was written to standardize the superimposition of two antibody structures. This script takes as input two antibody structure files: a mobile file and a reference file. The mobile file contains the structure that will be fitted onto the reference structure. These files are numbered using the kabatPDB.pl script unless they are already numbered (as in the case of the preprocessed high resolution set). From here common framework Kabat numbers are used as the

zones for superimposition. More specifically, the alpha carbons for each of these common residue positions are used for the superimposition. This system prevents the structurally ambiguous CDR residues from skewing the fit. The output from this program is the mobile file's coordinates and annotation information as well as the RMSD calculation for these two structures. This script was used heavily in dataset processing where all the high resolution structures were superimposed to a common structure to normalize their 3-dimensional orientation. This script was also used as the evaluation method in the benchmarking experiments that will be described in the next section.

The runmodeler.pl script combines all the pieces into a model generating pipeline. The runmodeler.pl script takes as input a chain sequence or pair of sequences and several parameters.

```
Usage: runmodeler.pl -h <heavy chain> -l <light chain> -s <scoring scheme>

Options:
-c: Print the results in the current directory
-outpath <path>: Print the results in a specified directory
-exclude <list of pdbs eg. 1abc,1abd,1abe>: Don't use these pdbs
-template <pdb code>: Specify the template to use
-weight <number>: Weight for non-homology position specific scores
-aligncdr: Center any gaps between template and query
-chnweight <chain,weight>: Use this to vary the weight of your
                           input chains. ie: -chnweight l,4 h,5

-M <matrix>: Use a particular substitution matrix
   Matrices: (PAM30, PAM70, BLOSUM45, BLOSUM62, BLOSUM80)

Scoring schemes:
A: Simple BLAST ranking
B: Sequence Homology
C: CDR length bonus
D: Heavy/Light Chain Interaction Bonus
E: Distance from to nearest CDR bonus
F: Solvent Accessibility
```

Figure 11: A direct copy of the usage statement printed out by the latest version of the runmodeler.pl script is shown here. It shows how the program is executed and what command line options are available.

The usage statement from the runmodeler.pl script is shown in figure 11. The parameters shown here help customize the output, and adjust what scoring functions will be utilized and what weights will be applied to them. Of particular interest are the –weight and –chnweight parameters. The –weight parameter allows for the customization of each scoring function score. If

a value of 5 is used as a weight then each score assigned from that scoring function will be multiplied by 5. Additionally, the –chnweight option allows for a similar functionality for each chain. If a chain weight of 5 is used then all scores for a given chain are multiplied by 5. This option was implemented to determine whether one chain's score was more important than another'sin terms of the overall quality of a two chain model. Another parameter that was particularly useful was the –exclude option. This option allowed for the high resolution set to be used as the test set since the –exclude could simply be used with the query's PDB code. It could also be useful to skip over one template and allow the next best one to be used. While being the workhorse script of the system, the runmodeler.pl script actually has very few parts and is quite easy to follow. Compartmentalization strategies allow the script to have a less cluttered appearance and make further improvements simpler.

Evaluation/Testing:

Some of the most difficult problems encountered in this project were the creation of a battery of tests and the integration of an evaluation method to test the quality of the models being generated. In particular, difficulty was encountered finding a method that could be used to calculate the quality of the models without being swayed by the abnormalities that could occur in the structure of the CDR regions. Several methods were tried including **F**lexible structure **A**lignmen**T** by **C**haining **A**ligned fragment pairs allowing **T**wists or FATCAT (Ye and Godzik 2003), MaxSub (Siew *et al.* 2000), and LGA (Zemla 2003). All of these methods are limited to one chain at a time and therefore would not be suitable for examining the quality of chain pair models. Despite this setback it was considered that these methods could provide insight into the quality of individual chain models. However, the problem found most often was a lack of sensitivity overall, and oversensitivity in certain areas. Most of these methods had disappointingly low variances when model results were compared across template selection methods. All of these methods utilized variations on the Root Mean Squared Deviation calculation. This calculation is

an accepted measure of the quality of a model when compared to its known structure. While RMSD is used as a basis for the methods listed above, these methods go further to incorporate other factors and report an overall score instead of a simple RMSD value in Å. While this was intriguing, it often muted the sensitivity of the method. Additionally, several outlier scores were often found because of oversensitivity at the terminal ends of a structure.

As a result, the method chosen was ProFit from Andrew Martin's group at UCL. While some bugs were found in the programming of ProFit, it provided an honest RMSD calculation not terribly different from the other methods and also awarded two important advantages: the ability to handle multiple chains and extraordinary flexibility in designating atoms for use in the calculation.

As was mentioned earlier, one problem that resulted from a standard RMSD calculation like the one from ProFit was particular sensitivity at the terminal regions. As figure 12 shows, the first and last 2 or 3 residues can vary considerably in their conformation without affecting the rest of the model. Unfortunately, when this variation occurs it can dramatically affect the RMSD calculation and give a poor score to a model that could potentially be of high quality. As a result, a pair of command line options was added to the superimposition wrapper program runprofit.pl: –pre and –post. –pre n ignores the first n residues and –post n ignores the last n residues. For the sake of evaluating the quality of the test models –pre 3 and –post 3 were used. This eliminated the oversensitivity problem as demonstrated in the figure 12 example. The results of this change are summarized in the first portion of the results section and the implications are described in more depth in the discussion section that follows.

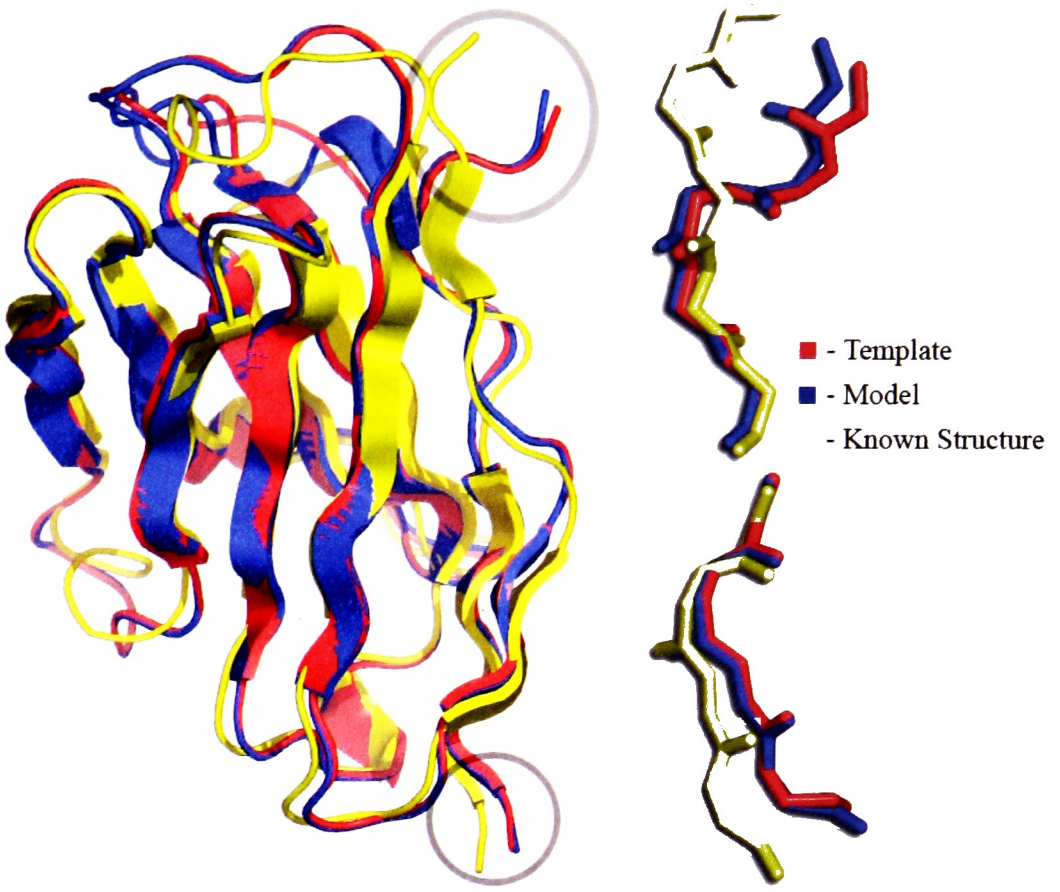


Figure 12: The sensitivity that occurs at the N and C termini is shown here. The circled regions are blown up to the right of the full size structure to show where slight differences in structure occur. The N-terminus is on top and the C-terminus is on the bottom. The template, model and known structure are all shown superimposed on each other. In this example the model is of 1ngzB and the RMSD calculation from ProFit is 1.092 Å when these residues are included and 0.648 Å when these residues are excluded using the pre/post option.

Standard Kabat Numbering

```
GFSLSTSGIG--
GFSLSTSGIGVT
```

Aligncdr Method

```
GFSLS--TSGIG
GFSLSTSGIGVT
```

Figure 13: The standard Kabat numbering method for CDRs and the AlignCDR method, which shifts the gaps to the center, are shown here.

Another adjustment, known as aligncdr, was made at the alignment step. Before this change was made, CDRs were numbered according to the strict Kabat rules in which the first residue of the CDR is the first number for that CDR region and the numbers increase till the CDR is complete. If there are extra residues then letters are added beginning with a. In other words, any gap is always added to the end of the CDR region. This method is shown pictorially in figure 13. However, a structural anomaly was found in several structures where

the CDR lengths between the template and query were different. This abnormality is shown in figure 14. In these cases, it was found that the MODELLER algorithm had difficulty dealing with gaps located at the ends of loops. As a result, the alignCDR option was implemented in which gaps typically located at the end of CDRs are shifted to the center of the CDR. This method is shown in figure 13.

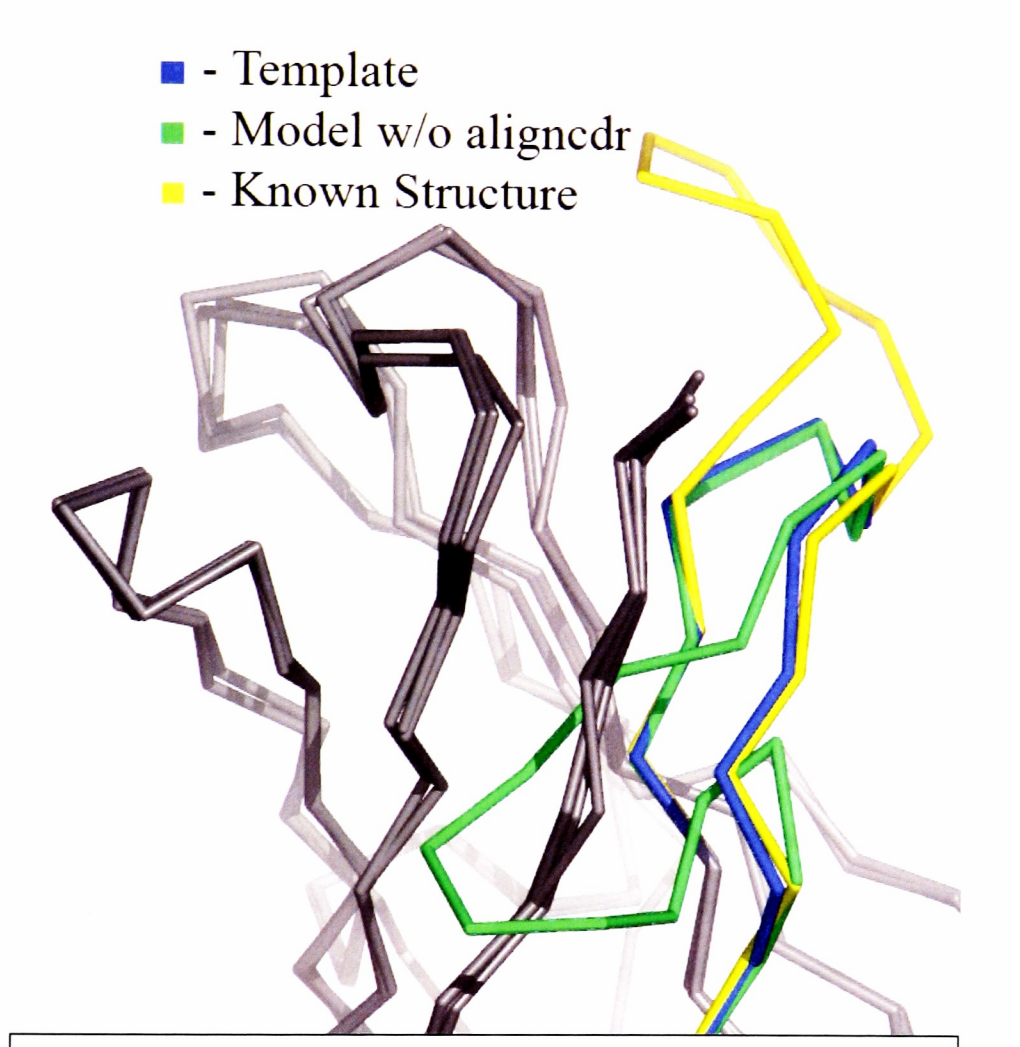


Figure 14: This is a molecular visualization shows 3 structures: the template, the model and the known structure for the query. As this visualization shows, there is a knot in the CDR3 region which ties around Framework 1. The model shown here is of PDB entry 1ospH when modeled with template 1gpoI. The RMSD of this model was 0.777 Å.

MODELLER seems to be trying to push these extra residues out of the loop structure when, in fact, they should be part of the loop. The theory of the aligncdr method was to force MODELLER

to include all of the CDR gap residues in the loop structure. The implications of this modification are summarized in the results section and are examined more fully in the discussion section.

In order to train the system and evaluate its effectiveness, several test runs were performed using a wide variety of parameters. To begin with, a small set of structures were removed from the high resolution set and were thus excluded from being chosen as templates. It became apparent, however, that this was not the most efficient way to perform the tests. By excluding the test set structures from the high resolution set, this setup introduced a potential for considerable bias and ambiguity. As a result, the setup was revised such that the entire high resolution set became the training set. The structure being queried with was simply excluded from the list of possible templates. This system allowed for the generation of much larger result sets and eliminated the problem of having ambiguously removed the best template from the high resolution set for use in the test set. The output reported in the results section will thus be representative of test runs that utilized the entire high resolution set as the test set. While several different configurations were tried, it became clear that trial and error experiments such as this would be limited in their ability to locate the best combination of scoring functions and weights.

As a result of the above observation, a brute force all vs. all experiment was started very near the end of the project's timeline. This experiment had the potential to create a model for every query/template pair in the entire high resolution library. This system would then calculate PIDs and scores for both individual chains and chain pairs and represent all the information that could be gathered from this set of training data. The numbers gathered from this one week run included the scores for each individual scoring function for all of the query/template pairs. A default weight of 1 was applied to all scoring functions with the exception of score function C which assigned a bonus of 10 for perfectly matched CDR lengths and 7.5 was used as a bonus for templates with longer CDRs than their queries. All told each entry in the high resolution set was

modeled 80 times, once for each of the remaining 80 structures in the library. This generated 6,480 structures in total. The amount of score data generated is daunting and will simply be summarized in this report.

Results:

This section will summarize some of the data generated over the course of the project. In particular the effects of the pre/post option, aligncdr option, and use of a substitution matrix will be explored. Additionally, data generated during the extensive training phase of the project will be summarized. Finally, the results of the all vs. all modeling experiment will be reported. Analysis of all of these data should be considered preliminary as the volume is prohibitive in terms of complete statistical analysis within the scope of this project. The discussion section that follows will describe in more detail some of the potential future projects that these data suggest, and how the information could be used to further enhance the system as a whole.

Pre/Post:

The effects of removing the first and last 3 residues from the RMSD calculation will be summarized with a box plot, descriptive statistics and hypothesis testing. Figure 15 shows a variety of summary statistics for the pre/post data sets. The standard descriptive statistics from MINITAB (Minitab Inc. 2003) are shown which include the mean, median and standard deviation. This section also includes the number of data points in the set, which shows that 335 query template pairs are represented. Also included in this summary are two histograms, one for the set with the pre/post option and one for the set without the pre/post option. Finally a boxplot is included showing the distribution of scores taken from each set.

Descriptive Statistics: with pre/post, without pre/post

Variable	N	N*	Mean	SE Mean	StDev	Minimum	Q1	Median
with pre/post	335	0	0.6719	0.0121	0.2208	0.2890	0.5060	0.6320
without pre/post	335	0	0.7876	0.0159	0.2907	0.3330	0.5520	0.6970

Variable	Q3	Maximum
with pre/post	0.7860	1.3780
without pre/post	0.9830	1.6520

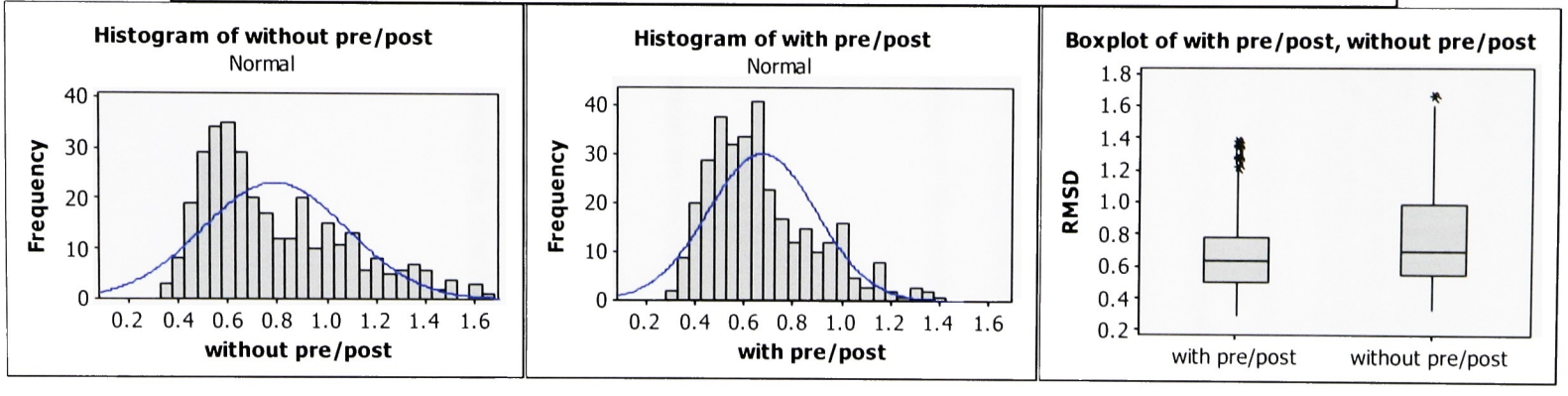


Figure 15: This is the summary information for the training data used to investigate the affect of the pre/post option on RMSD. The Descriptive Statistics from MINITAB are shown on top. Next are two histograms, one for the set with the pre/post option and one for the set without the pre/post option. Lastly is a boxplot which shows the distribution for both of the data sets in this experiment.

Sign Test for Median: DiffPrePost

```
Sign test of median    0.00000 versus not = 0.00000

              N  Below  Equal  Above       P    Median
DiffPrePost 335    302      3     30  0.0000  -0.04000
```

Figure 16: These are the results from the Sign Test as calculated by MINITAB. As the information shows, the p-value of 0 means that the null hypothesis, the two sets have equal means, can be rejected and the alternative, that the two have significantly different means, can accepted.

Figure 16 contains the results of a sign test when performed on the differences between the with pre/post data set and the without pre/post data set. In this experiment the null hypothesis, H_{o}, was that the medians of the two sets were equal, while the alternative hypothesis was that the two medians were not equal. The result is a p-value of 0 which is less than the alpha of 0.05. Therefore the null hypothesis can be rejected and the alternative hypothesis can be accepted with a confidence of 95%. The implications of this result will be investigated more fully in the discussion section.

Aligncdr:

The results of the aligncdr experiment are summarized in figure 17. This box contains the same information found in the description of the pre/post experiment above. Descriptive statistics from MINITAB, histograms for both the with aligncdr option set and the without aligncdr option set are present along with boxplots showing the distribution for both sets.

Descriptive Statistics: Without Aligncdr, With Aligncdr

Variable	N	N*	Mean	SE Mean	StDev	Minimum	Q1	Median
Without Aligncdr	248	0	0.6725	0.0146	0.2299	0.3230	0.5120	0.6320
With Aligncdr	248	0	0.6547	0.0131	0.2070	0.2890	0.4998	0.6135

Variable	Q3	Maximum
Without Aligncdr	0.7868	2.2720
With Aligncdr	0.7803	1.2820

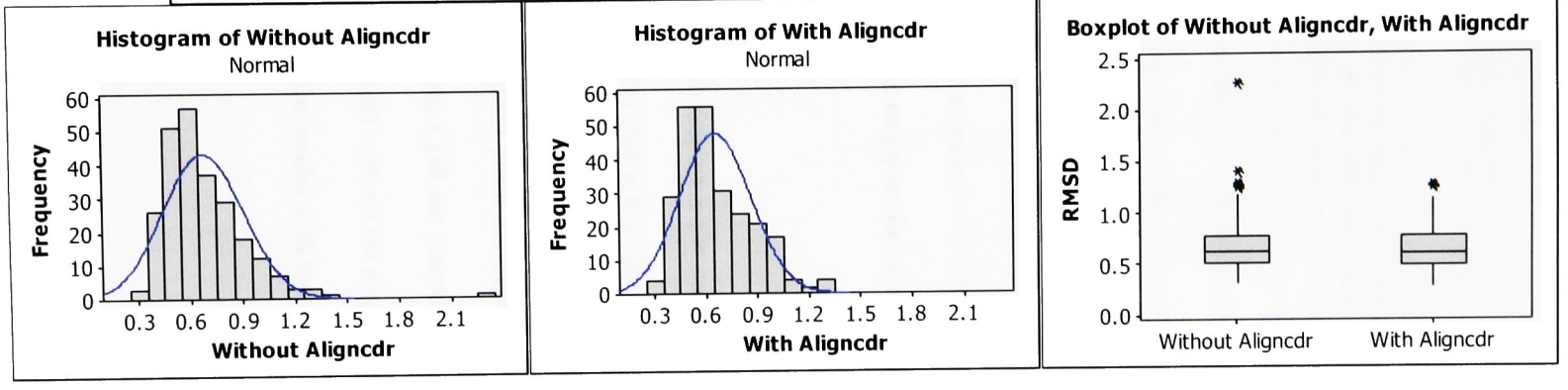


Figure 17: This is the summary information for the training data used to investigate the affect of the aligncdr option on RMSD. The Descriptive Statistics from MINITAB are shown on top. Next are two histograms, one for the set with the aligncdr option and one for the set without the aligncdr option. Lastly is a boxplot which shows the distribution for both of the data sets in this experiment.

Sign Test for Median: DiffAlgn

```
Sign test of median =  0.00000 versus not = 0.00000

            N  Below  Equal  Above       P    Median
DiffAlgn  248    125     70     53  0.0000  -0.00100
```

Figure 18: These are the results from the Sign test as calculated by MINITAB. The p-value of 0 indicates that the null hypothesis, the two sets have equal medians, can be rejected and the alternative, that the two have significantly different medians, can be accepted with more than 95% confidence.

Figure 18 shows the results from the sign test as calculated by MINTAB. The null hypothesis for this experiment, H_o, is that the medians of the two sets are equivalent. Likewise the alternative hypothesis, H_a, is that the two medians are not equivalent. Like the test for pre/post, this experiment has a p-value below the alpha of 0.05 and therefore the null hypothesis can be rejected and the alternative can be accepted with a confidence level of 95%. As in the pre/post experiment, the difference is in the negative direction suggesting that the aligncdr option lowered the median RMSD score.

In addition to the significance with respect to the median RMSD, the aligncdr option makes a difference in terms of model quality that is most probably overlooked by the RMSD calculation. This difference is illustrated in figure 19. In this figure one can see how the aligncdr option has helped MODELLER to more properly place the CDR loop and avoid the knot problem encountered when the gaps from the CDR are placed at the end. In this visualization the template structure is shown in blue, the model without the aligncdr option is shown in green, the known structure is shown in yellow and the model with the aligncdr option is shown in red.

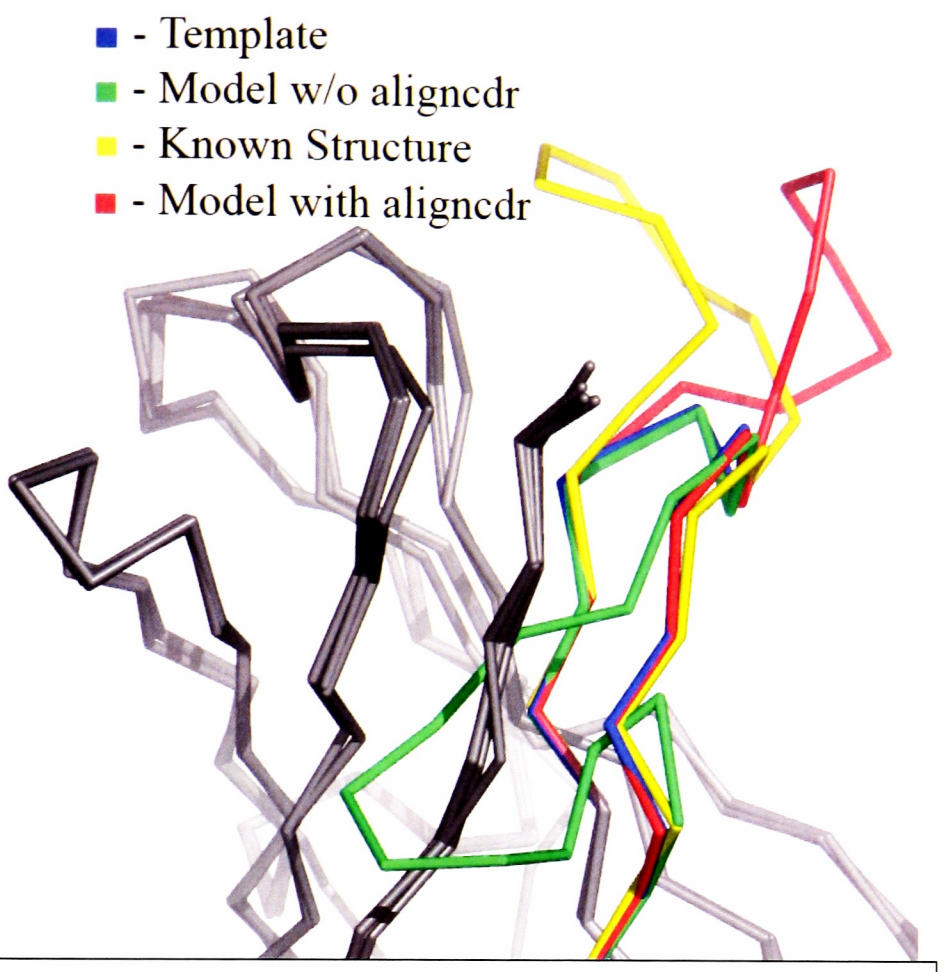


Figure 19: This is a molecular visualization showing how the aligncdr method fixes the knot problem encountered in 1ospH. While this fix causes a significant change in the CDR conformation, it does not necessarily have a deleterious effect on the framework structure. In the case of 1ospH the RMSD for the model without aligncdr is 0.777 and the RMSD with aligncdr is 0.513.

Matrix:

As mentioned earlier, the template selection method can be run using a substitution matrix. One question that was asked was whether or not using a substitution matrix made a significant difference on selection. Figure 20 shows the histograms and boxplots of the results with and without a matrix (in this case the BLOSUM 62 matrix).

Descriptive Statistics: IDENTITY, MATRIX

Variable	N	N*	Mean	SE Mean	StDev	Minimum	Q1	Median	Q3
IDENTITY	56	0	0.62591	0.00207	0.01550	0.60306	0.61554	0.62248	0.63394
MATRIX	66	0	0.62731	0.00226	0.01836	0.60900	0.61514	0.61956	0.63450

Variable	Maximum
IDENTITY	0.67570
MATRIX	0.67677

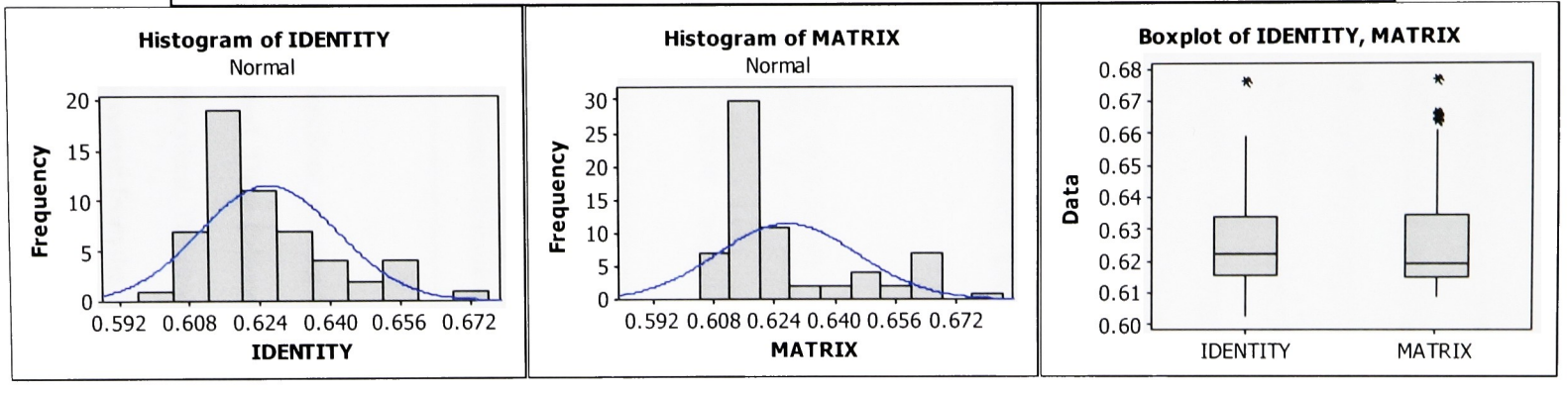


Figure 20: This is the summary information for the training data used to investigate the affect of a substitution matrix on RMSD. The Descriptive Statistics from MINITAB are shown on top. Next are two histograms, one for the set with the matrix and one for the set without the matrix. Lastly is a boxplot which shows the distribution for both of the data sets in this experiment.

The box in figure 21 shows the results of a Mann-Whitney test between selection performed with a matrix and selection performed without a matrix. The Mann-Whitney test was used here since the two samples could not be considered paired. In addition to adding the matrix, the weights for each scoring function were typically increased to compensate for the increase in overall score provided by the matrix. As a result there were more tests runs. In this case the null hypothesis, H_o, is that the two samples have the same medians, while the alternative hypothesis, H_a, states that they have a significant difference in their median values. In this case, with a p-value of 0.8291, when corrected for ties, there is insufficient evidence to reject the null hypothesis. Therefore this data shows no significant difference between the medians of the two sets.

Mann-Whitney Test and CI: IDENTITY, MATRIX

```
           N   Median
IDENTITY  56  0.62248
MATRIX    66  0.61956


Point estimate for ETA1-ETA2 is 0.00048
95.0 Percent CI for ETA1-ETA2 is (-0.00389,0.00468)
W = 3486.5
Test of ETA1 = ETA2 vs ETA1 not = ETA2 is significant at 0.8292
The test is significant at 0.8291 (adjusted for ties)
```

Figure 21: The are the results of the Mann-Whitney test for equality of medians between test results generated with a matrix and those generated without a matrix. The results show that there is not a significant difference between the medians of these two sets. This is evidence by the high p-value (0.8291 when corrected for ties). As a result, one has insufficient evidence to reject the null hypothesis and must accept that there is no significant difference between these two sets.

Training:

Several combinations of scoring functions, scoring function weights and chain weights were tried in search of the best selection method. These different combinations were run on the entire high resolution set, as was previously described. The results from these tests were compiled and ranked using a Friedman ranksum system from the XLSTAT package (XLSTAT 2006). The best combinations are shown below in table 2.

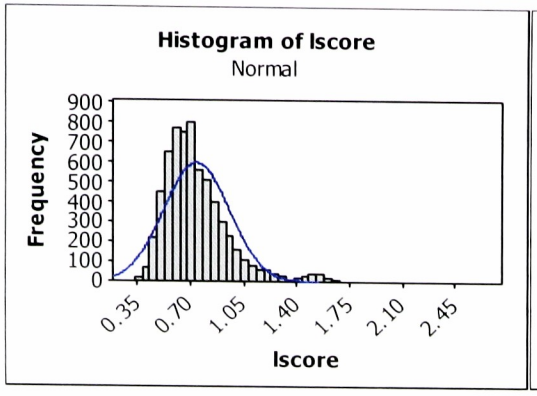
Sample	Frequency	Sum of ranks
BF_M_BLOSUM62_weight_1_aligncdr_015	81	374818.500
BC_weight_3_aligncdr_100and75	81	377636.500
BF_M_BLOSUM62_weight_5_aligncdr	81	377999.500
BC_weight_3_aligncdr_100and100	81	378860.000
BF_M_BLOSUM62_weight_5_aligncdr_012	81	379273.500
BC_weight_5_aligncdr_100and50	81	379920.500
BF_weight_3_aligncdr	81	381452.000
BF_M_BLOSUM62_weight_6_aligncdr	81	382126.000
BC_weight_5_aligncdr_100and75	81	382386.000
BC_weight_5_aligncdr	81	384007.000

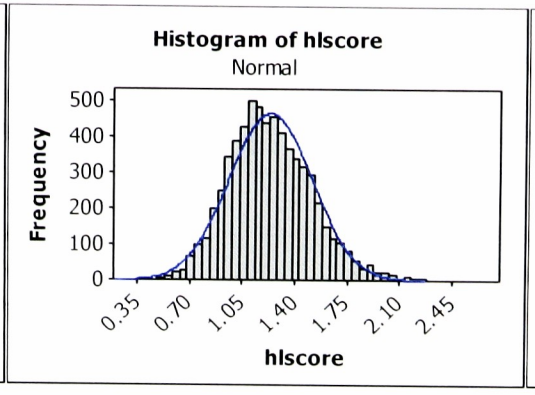
Table 2: Here are the top 10 score function combinations for the heavy set training. Decoded the first one reads: score functions B and F with BLOSUM62 matrix, a weight of 1 for the F function, aligncdr used for CDR alignment, and a binning system of 0.1 and 5 for buried residues. Some combinations involving score function C include a ratio 100and75 this indicates the percent bonus given to template CDRs of equal length and template CDRs of greater length respectively.

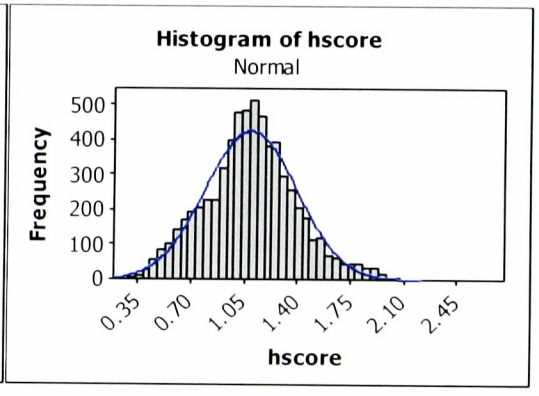
As will be discussed later, the issue with this ranksum is that, while this measurement is sensitive enough to rank these methods, the differences between the RMSD sets for each score function combination are not statistically significant (with the exception of the few worst ranked and few best ranked). This issue speaks volumes about the data set, and these concerns will be explained more fully in the discussion section.

All vs. All:

This analysis took 7 days and generated 6,480 3-D models, one for every query/template combination in the high-resolution set. In addition to the modeling step, each query/template pair was scored based on the default parameters for each of the 6 scoring functions. These scores were compiled, along with the RMSD calculations and the percent identity calculated for each query/template. The RMSD and PID were collected in 3 forms: heavy chain only, light chain only and paired chains. The RMSD scores are summarized in the three histograms in Figure 22. The boxplot illustrates the difference in distributions between the heavy chain scores and the light chain scores and the two correlation matrices show the relationship between percent identity, single chain score and combined chains score.

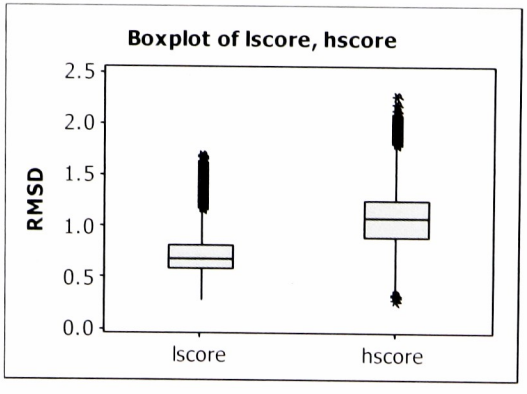






Correlations: lpid, lscore, hlpid, hlscore

	lpid	lscore	hlpid
lscore	-0.225		
	0.000		
hlpid	0.653	-0.162	
	0.000	0.000	
hlscore	-0.130	0.428	-0.307
	0.000	0.000	0.000



Correlations: hpid, hscore, hlpid, hlscore

	hpid	hscore	hlpid
hscore	-0.514		
	0.000		
hlpid	0.838	-0.420	
	0.000	0.000	
hlscore	-0.307	0.611	-0.307
	0.000	0.000	0.000

Figure 22: These data come from the all vs. all experiment. The histograms on the top from left to right show distributions for the light chain, combined chain and heavy chain RMSD scores respectively. On the bottom left and right are two small correlation matrices showing the relationships between percent identity, single chain score and combined chain score for light chain on the left and heavy chain on the right. In the middle at the bottom is a boxplot of the light chain score and heavy chain score plotted on the same axis.

The results were further analyzed by generating correlation matrices in MINITAB using the results from the different scoring functions. These matrices are shown in Tables 3 and 4. Table 3 shows the correlation matrix using the heavy chain results while table 4 shows the correlation matrix using the light chain results. There are two values presented in these tables for each correlation. The top value is the Pearson Correlation Coefficient and the bottom value is the p-value representing the confidence in that correlation.

	Heavy_A	Heavy_B	Heavy_C	Heavy_D	Heavy_E	Heavy_F	hscore
Heavy_B	0.928						
	0						
Heavy_C	0.391	0.349					
	0	0					
Heavy_D	0.433	0.442	0.211				
	0	0	0				
Heavy_E	0.48	0.484	0.185	0.129			
	0	0	0	0			
Heavy_F	0.779	0.882	0.258	0.371	0.466		
	0	0	0	0	0		
hscore	-0.506	-0.53	-0.145	-0.124	-0.33	-0.484	
	0	0	0	0	0	0	
hlscore	-0.32	-0.321	-0.055	-0.131	-0.169	-0.307	0.611
	0	0	0	0	0	0	0

Table 3: The correlation matrix generated using the heavy chain all vs. all data is shown here. The values reported are the Pearson Correlation Coefficient and the p-value respectively. In this case the 6 different scoring functions were compared with themselves as well as the heavy chain RMSD scores and the combined chain hlscore.

	Light_A	Light_B	Light_C	Light_D	Light_E	Light_F	lscore
Light_B	0.893 0						
Light_C	0.275 0	0.152 0					
Light_D	0.512 0	0.623 0	-0.057 0				
Light_E	0.276 0	0.26 0	-0.003 0.797	0.151 0			
Light_F	0.558 0	0.631 0	0.065 0	0.584 0	0.01 0.402		
lscore	-0.284 0	-0.26 0	0.013 0.299	-0.211 0	-0.266 0	-0.151 0	
hlscore	-0.145 0	-0.156 0	0.017 0.164	-0.208 0	-0.185 0	-0.113 0	0.428 0

Table 4: The correlation matrix generated using the light chain all vs. all data is shown here. The values reported are the Pearson Correlation Coefficient and the p-value respectively. In this case the 6 different scoring functions were compared with themselves as well as the light chain RMSD scores and the combined chain hlscore.

Discussion:

This section will more fully describe our interpretation of the results presented in the previous section. Reference will be made to figures and tables from the preceding section, but whenever practical the numbers pertinent to the analysis will be duplicated for clarity. The topics analyzed here include: the pre/post option, aligncdr option, substitution matrix, training data/ranksum of scoring functions and all vs. all correlation values from the Results section. In addition to these topics, an overview of the currently available alternatives and suggestions for future expansion of this work will conclude this section.

Pre/Post:

This experiment revealed information about the RMSD calculation method as well as the distribution of the model scores. The descriptive statistics and histograms from figure 15 show that the data are not quite normally distributed, rather they are skewed to the right whether the pre/post option is used or not. The plots show that the modeling favors scores between 0.4 and 0.8 Å RMSD with more scores above this range than below it. When the pre/post option was added, outlier scores above 1.4 were eliminated. These higher RMSD models are those instances where the ProFit program was influenced heavily by the structurally polymorphic N and C terminals. The boxplot clearly shows that the pre/post distribution is more tightly centered on the median than the without pre/post option. In addition the pre/post option causes scores above 1.2 Å to become statistical outliers where they were previously considered normal.

The results of the sign test give statistical significance to the per/post method. The sign test result from figure 16 shows, with high confidence, that the pre/post option shrunk the median by 0.04 Å. This negative value indicates that the terminal ends were indeed causing an oversensitivity leading to higher than necessary RMSD calculations. While finding the differences that cause one

template to be better than another is the purpose of this system, it cannot have over sensitivities such as these cluttering the datascape. A model given a score of 1.4 Å instead of 1 Å based on the structural variability of only 6 of 70 residues is not acceptable. As a result, this option was employed for all remaining training runs as well as the all vs. all experiment.

Aligncdr:

The aligncdr option data illustrate the susceptibility of the modeling protocol to error in the presence of a length polymorphism. Figure 19 shows how the position of gaps in the alignment can drastically change the final model. Additionally, the descriptive statistics and histograms in figure 17 show the removal of a few very high RMSD models and also the redistribution of models between 0.4 and 1.0 Å. This is significant since it indicates that the position of CDR gaps can have a dramatic effect on the RMSD calculation, potentially spoiling a model of high quality. The Sign Test results shown in figure 18 confirm that the addition of the aligncdr option lowered the median RMSD value. While this system is designed specifically for framework modeling onto which a CDR modeling tool could be applied, an error of this magnitude must be corrected at this level for a quality model to be realized in the end. In addition, abnormalities such as these do not have predictable effects on framework RMSD. In some cases the errant CDR loop will displace some framework residues and will result in a high RMSD score. These cases can be detected. However, in other cases the displaced loop may manage to wrap itself around a framework and not cause a significant increase in framework RMSD. These immeasurable effects are also deleterious since these models contain a major structural error but appear to be normal.

One potential issue with the aligncdr option is that it somewhat overlaps the functionality provided by the score function C, which gives bonuses for template CDR lengths greater or equal to the length of the query CDR. While this is the case, the value of score function C is in determining if the length of a CDR has, in itself, any inherent effect on the structure of the

framework. Score function C is not designed to improve the ability of MODELLER to model frameworks, it is simply to investigate the relationship between CDR length and overall framework structure. An example of a potential question that score function C asks is, "do chains with CDRH1's of length 4 typically have the same framework structure leading up to the CDR?" This is a very different idea than the issue addressed by the aligncdr option, which is simply to level the playing field between templates whose CDR length polymorphisms cause an RMSD increase and those that do not.

Matrix:

The matrix option was seen originally as a necessity, improving the sensitivity of the system. It was thought that there was no way to achieve sensitivity comparable to BLAST without using a substitution matrix at least with the homology score function B. However, several tests were run using a matrix during the training phase and it appears as though the matrix is not as important as was originally thought. The results of these tests are shown in figure 20. Here the descriptive statistics and histograms show a non-normal distribution, we see a slight rightward skew as was seen in the pre/post and aligncdr. There is very little difference apparent between the two data sets when the mean and median are compared. In addition, figure 21 shows the results of a Mann-Whitney test in which a high p-value indicates insufficient evidence to suggest that these two sets are actually different.

One thing to note about the matrix results is that the two sets are of different sizes, 56 entries in the identity set and 66 entries in the matrix set. The reason for this difference is that several more weight combinations were tried with the matrix. The matrix turned a discrete identity based scoring function into a continuous homology based scoring function. This change required that the weights for scoring function combinations be adjusted to take into account the much higher overall homology score. Weights from 1-10 were typically used for combining scoring functions

in the identity system, but these values were increased to 10-100 for combination of the homology based system. As a result, the non-paired nature of the Mann-Whitney test was necessary to determine if the matrix made a significant effect.

The fact that the matrix does not improve the selection method was one of the early indications that the structural data set was not particularly diverse. This issue will be discussed more fully in a later section, but it is important to note that a substitution matrix is typically a valid way to improve the sensitivity of a search/scoring method In this case it was ineffective. This point speaks to the fact that the structure data being used should not be considered a random sample from a diverse population. The population being searched is very focused and the difference of only a few amino acid positions can determine whether a template is chosen or not. Matrix scores are most effective in homology detection. The problem being addressed here is the selection of the closest homolog from a set of homologs. As a result a different matrix with different underlying principles may be required. One must also remember that the matrix is a selection-time adjustment directly linked to what template is selected. This is fundamentally different than either the pre/post or the aligncdr which are score-time and align-time adjustments. These results suggest that the use of a matrix is an unnecessary complication in the scoring functions, and that simple residue identity is sufficient to distinguish between the structures in the high-resolution set.

Training:

The results of the training provided insight into the nature of the structure data set, and gave clues to the possible limitations of the method. The training was performed using score functions A and B alone as well as B paired with C, D, E and F. The top 10 results are shown in table 2 and reveal that the BF and BC combinations were best for the heavy chain. This table also shows how the

matrix results seem intermixed with the nonmatrix results. This further reinforces the conclusion made in the last section regarding the ineffectiveness of the substitution matrix.

This ranksum indicates that the two scoring function combinations that were most successful were BF and BC. Score function F gives bonuses for buried residues that match while score function C gives bonuses for template CDRs that are equal to or greater in length than the corresponding query CDRs. These data suggest that a bonus of approximately 5 for each matching buried residue is quite successful when using a matrix. A slightly smaller weight of 3 is more appropriate when not using a matrix. For score function C it appears that a bonus of 3 or 5 is best when using identity. It also appears that if the length is greater the bonus can be reduced to 75% of the total to slightly improve results. It is promising to see that score function combinations such as these improve on the score function B based solely on sequence identity/homology. In addition these results are all ranked much higher than the results generated using score function A, based on the BLAST e-value.

Despite the promise of improved template selection provided by this ranksum, there is unfortunately no statistically significant difference between any of these score function combinations and BLAST. When a Sign Test is performed between even the best score function combination and the score function A results there was no significant difference between the two result sets. This fact suggests that there may be some limitations in the data set with respect to the divergence of the sequences and structures. Appendix 3 shows a very large boxplot containing the distributions for all of the heavy chain training data. While some variability is present, it is very difficult to identify trends or suggest that one or a group of score function combinations is the strongest. The fact remains that of the 81 structures in the high-resolution training set, a considerable portion of them result in the same template selected and therefore the same RMSD

score. This limitation may be a product of too small a data set and/or not enough diversity in this set.

All vs. All:

The results of the training activities suggested that a new strategy be adapted for optimizing the scoring functions and exploring the potential limitations of the data sets. This strategy was the all vs. all experiment discussed in the materials and methods section. One of the byproducts of the all vs. all experiment was the ability to see what the best possible template selection score could be for each query. The idea is that this result could provide insight into whether or not further exploration into improving the current method is worth while, or whether the data set is truly insufficient for this type of analysis.

The distributions shown in figure 22 will help to determine whether or not the data in the high resolution set are sufficient. Figure 22 shows the distributions for each chain, as well as the combined chains. One thing that is quite obvious is that the heavy chain seems to be more normally distributed than the light chain, which is skewed to the right. This indicates that the light chain models tend to favor a low RMSD score, and do not as regularly have scores above approximately 1 Å. This is in contrast to the heavy chain, which has a mean RMSD score near 1 Å. The combined score seems somewhat of a compromise in that it is more normally distributed than the light, but does seem to favor lower RMSD scores slightly. These results suggest that perhaps the light chain data is not as diverse and/or not as difficult to select a template for.

The correlations between PID, chain score and combined chain score shed even more light on the disparity between the heavy and light chain data sets. While all of the correlations shown in both matrices are significant, with p-values of 0, the Pearson coefficient values indicate the strength of the correlation and show that PID is more than twice as strongly correlated with RMSD in heavy

chains than in light chains. The values are 0.514 for heavy chain and 0.225 for light chain. This suggests that sequence homology is very important in selecting a template for heavy chain modeling, but only half as important in light chain modeling. While one may construe this as indicating that the light chain selection is actually *more* complicated, because PID is not enough to select a good template, when one combines this information with the distribution shown in the histogram in Figure 22 it becomes clearer that the light chain data are simply less divergent and therefore it is actually easier to select a template. The boxplot in figure 22 supports this hypothesis in that if one randomly selected a score from the light chain set it would have a good chance of being lower than one selected randomly from the heavy chain set. In fact, this phenomenon was suspected early on in the training experiments and is one reason that light chain training was stopped and focus was placed on the heavy chain.

Range of RMSD scores:

Chain: Best ------Worst
Heavy: 0.5367 ------ 1.8392
Light: 0.4738 ------ 1.5468

Figure 23: The range of possible RMSD values derived from the all vs. all analysis are shown here.

While these distributions are interesting in the context of describing the data sets, they do not address the issue of what the optimum RMSD score can be, or what the range of possible RMSD values are for the set as a whole. For this the best and worst set of RMSD scores (one score for each query) for the heavy chain and light chain were retrieved from the results and examined. Figure 23 shows the possible range of values for the heavy and light chains. These ranges demonstrate the best and worst possible template selection results. The best template selection method tried during the training described in the last section had an average RMSD of 0.609 Å. BLAST gave an average RMSD of 0.643 Å and score function B alone gave 0.626 Å. These values are all near the lower end of the heavy chain range, however there is some room for improvement. A sub 0.6 prediction average for this training set would be an attainable goal if the scoring function could be adjusted properly. These adjustments, however, are complex since so many variables are involved.

In order to deal with the complexity involved in combining the scoring functions and weights, the all vs. all query/template pairs were scored with each scoring function. These scores were collected so that they could be used to find the best combination of scores. The hope was to examine how each score function contributes to the RMSD. As part of the analysis of these data the correlation matrices found in Tables 3 and 4 were generated. These matrices show how highly correlated each scoring function's score is with the RMSD score for both the individual chain and also the combined chains. In addition, these matrices indicate the level of relatedness between the different scoring functions.

Table 3 shows the heavy chain correlation information. As one would expect, scoring function B is highly correlated to A. This is because A is the BLAST e-value and B is the identity score. Both of these scores are based on sequence similarity, however B does not include CDR regions in its calculation. Score function C is not well correlated with any of the other scoring functions since it is the only one that is not sequence identity based. Score function D and E show a similar pattern of being only moderately correlated with any of the other scoring functions. However score function F seems to be strongly correlated to score functions A and B and then poorly correlated with the others. This phenomenon could indicate that the buried residues, those emphasized by score function F, are a footprint of sorts that is directly linked to overall sequence similarity. It could also point to the fact that overall sequence similarity and filtered sequence similarity, like that measured by score function F, are inherently related and therefore will have strong correlations. The score functions' correlation with RMSD in the individual chain shows that score function B is most highly correlated, followed by A, F, E, D and C respectively. This same trend is demonstrated in comparing the RMSD score to the combined chains. The last number on this correlation matrix is the correlation between the heavy chain RMSD and the combined chain RMSD. Here there is a fairly strong correlation 0.611 between the heavy chain

RMSD score and the combined. In this case, one would not do that badly using the heavy chain alone when selecting a structure template for both chains.

The light chain correlation paints a similar picture. The correlations between the different scoring functions follow nearly the same pattern as that found in the heavy chain. The few exceptions are E, C and E, F where no significant correlations were found. The major difference between the light chain results and the heavy chain results is that the correlations between the light chain score functions and the RMSD score are quite low in comparison to the heavy chain. This again harkens back to the conclusion drawn in the previous section regarding the light chain being easier to choose a template for. Additionally, the light chain RMSD score is not as highly correlated with the combined chain RMSD score. This is more evidence that the heavy chain is more variable than the light.

While these correlations help to reveal how the scoring functions contribute to the RMSD score, the correlations do not suggest how the scoring functions should be combined. This task is more complex than these simple correlations can describe. Further analysis of these results is necessary and will be discussed in the Future Enhancements section below.

Alternative methods:

To give this discussion some perspective this section will highlight some of the techniques currently available for automated template selection and antibody modeling. One of the leading automated methods is SWISS-MODEL hosted by ExPASY (Schwede *et al.* 2003). SWISS-MODEL has a mode known as "First Approach mode" in which a sequence can be input and the system will search its filtered set of templates known as ExPDB. This method utilizes a simple BLAST or PSI-BLAST search method to locate a suitable template structure. SWISS-MODEL

goes a step further, however, in that it superimposes up to 5 template structures, removing those with high RMSDs to the best template.

Another technique, this one specific to antibodies, is the Web Antibody Modeling system or WAM (Whitelegg and Rees 2000). The WAM algorithm is the most sophisticated system known for modeling antibodies. In this system the framework templates are chosen based solely on sequence homology. WAM does not consider the modeling of the framework to be particularly difficult. One reason for this is because WAM goes beyond this step and models the CDR regions separately based first on the canonical classes and second using a PDB search for a similar CDR. Several different models are generated in this system and the one with the lowest free energy is selected in the end. While WAM is by far the best automated method for modeling antibodies, it focuses primarily on the CDR regions since they are typically the most interesting to researchers. WAM represents what the system described herein could become if it were expanded to include a sophisticated algorithm for modeling CDR regions. Unfortunately the WAM system is currently unavailable as the host, at Bath University, is experiencing hardware problems.

A system similar to WAM was described by Morea and coworkers in which the framework and CDR regions are modeled separately (Morea *et al.* 2000). Like WAM, the framework regions are modeled with the template of highest sequence homology, unless a lower scoring template structure is of considerably higher quality. The alignment step is done in a fashion similar to the ClustalW system used here in that each sequence is aligned to a large number of sequences to identify where key residues are located. Once these regions have been identified, modeling of the frameworks begins. Each chain is modeled independently and the two chains are combined. This way each chain will have the greatest chance of having the best possible framework model. This method does not appear to have a computational implementation, however this protocol could be incorporated into the system described here. The framework modeling step is followed by a

sophisticated CDR modeling step similar to WAM. Like WAM, the focus of this method is accurate CDR modeling.

Future Enhancements:

This project has ended with many things accomplished and many new possibilities revealed. One aspect of the project that was cut short due to time constraints was the analysis of the all vs. all data. This data set includes which residues were identical for all the query/template pairs used. This means that these data could be used to analyze what residue positions are most important in template selection. A neural network could be implemented in which the inputs are the sequence identity footprints from the all vs. all data and the outputs are the RMSD scores. This type of system could show what residue positions need the most weight for the lowest possible RMSD. This type of system would suggest a new scoring function that optimizes the residues the network finds most important. An alternative neural network system could be implemented in which the inputs are the scores from each scoring function and the output is the RMSD score. This network would optimize the weights from the different scoring functions and suggest what physical/chemical properties are most important to framework modeling. The size and complexity of the all vs. all data lend themselves well to an analysis tool like the neural network.

Another future enhancement could include expansion to other molecule types. The system was designed with modularity in mind and could be expanded to accommodate a variety of molecules. The original plan for this project was to include expansion of the system to include the kinase protein. This expansion, as well as the expansion to other molecules, would separate this system from the currently available automated modeling solutions. In addition to expanding the system to different molecules, the addition of a sophisticated and automated CDR modeling routine like those described above would greatly enhance the functionality of the system. While this project

would be fairly involved, it would create a completely automated pipeline style modeling system like WAM.

As was suggested several times throughout this work, the structure data set is the key to the success of the system. This data set underwent several incarnations to arrive at its current state. Continued adjustment of the data set could undoubtedly produce different results and could potentially improve performance. As more divergent structure data is deposited the system can only get stronger.

Concluding remarks:

This project provided a unique experience in that it involved integrating different analysis programs into one cohesive pipeline. In addition it required working in an industry setting with an open ended research atmosphere and brilliant scientists. The project realized its goals while shedding light on some of the pitfalls and intriguing challenges associated with protein structure prediction. The author hopes that his efforts will benefit the organization he worked with and help further the field.

The author wishes to humbly thank the researchers J. Dutko, D. Sloth, A. Straw, M. Page, J. Snowden, P. Scordis, J. Bouck and, most of all, his primary advisor J. Shi whose Herculean efforts, brilliance and patience are behind every word in this manuscript.

References:

"Antibody Structure". 2000, University of Arizona. 21 Sept 2005. <http://www.biology.arizona.edu/immunology/tutorials/antibody/structure.html>.

Berman, H.M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T.N.; Weissig, H.; Shindyalov, I.N.; Bourne, P.E., "The Protein Data Bank". *Nucleic Acids Research*. 2000, 28:235-242.

Burley, S.K.; Almo, S.C.; Bonanno, J.B.; Capel, M.; Chance, M.R.; Gaasterland, T.; Lin, D.; Sali, A.; Studier, F.W.; Swaminathan, S., "Structural genomics: beyoned the human genome project". *Nat Genet*. 1999, 23(2):151-157.

Cheek, S; Ginalski, K.; Zhang, H.; Grishin, N.V., "A comprehensive update of the sequence and structure classification of kinases", *BMC Struct Biol*. 2005, 5:6.

Cozzetto, D.; Tramontano, A., "Relationship Between Multiple Sequence Alignments and Quality of Protein Comparative Models", *Proteins*. 2005, 58:151-157.

DeLano, W.L., "The PyMOL Molecular Graphics System" 2002, DeLano Scientific, San Carlos, CA, USA.

Eddy, S.R. "HMMER: profile HMMs for protein sequence analysis" 2003 Washington University St. Louis. 19 Sept 2005.<hmmer.wustl.edu>.

Eswar, N.; John, B.; Mirkovic, N.; Fiser, A.; Ilyin, V.A.; Pieper, U.; Stuart, A.C.; Marti- Renom, M.A.; Madhusudhan, M.S.; Yerkovich, B.; Sali, A., "Tools for comparative protein structure modeling and analysis", *Nucleic Acids Research*. 2003, 31(13):3375-3380.

Fiser, A., "Protein structure modeling in the proteomics era", *Expert Rev Proteomics*. 2004, 1(1):97-110.

Kay, L.E., "NMR studies of protein structure and dynamics", *J Magn Reson*. 2004, 173:193-207.

Lefranc M.-P.; Giudicelli V.; Kaas Q.; Duprat E.; Jabado-Michaloud J.; Scaviner D.; Ginestoux C.; Clément O.; Chaume D.; Lefranc G., "IMGT, the international ImMunoGeneTics information system®". *Nucl. Acids Res*. 2005, 33:D593-D597.

Martin, A.C.R.; "ProFit". *Dr. Andrew C.R. Martin's Group at UCL*. 2005. 19 Sept 2005 < http://www.bioinf.org.uk/software/profit/index.html>.

Mintab Inc. (2003). MINTAB Statistical Software, Release 14 for Windows, State College, Pennsylvania. MINTAB ® is a registered trademark of Minitab Inc.

Montelione, G.T.; Zheng, D.; Huang, Y.J.; Gunsalus, K.C.; Szyperski, T., "Protein NMR spectroscopy in structural genomics", *Nat Struct Biol*. 2000, Suppl:982-985.

Morea, V.; Lesk, A.M.; Tramontano, A., "Antibody Modeling: Implications for Engineering and Design", *Methods*. 2000, 20(3):267-79.

Pande, V.S., "Science of Folding@Home", *Folding@Home Distributed Computing*, 2005 Stanford University. 15 Sept 2005 <http://folding.stanford.edu/science.html>.

Pederson, J.T.; Henry, A.H.; Searle, S.J.; Guild, B.C.; Roguska, M.; Rees, A.R., "Comparison of Surface Accessible Residues in Human an Murine Immunoglobulin Fv Domains", *J. Mol. Biol.* 1994, 235:959-973.

Peitsch, M.C., "About the use of protein models", *Bioinformatics*. 2002, 18(7):934-938.

Petrey, D.; Xiang, Z.; Tang, C.L.; Xie, L.; Gimpelev, M.; Mitros, T.; Soto, C.S.; Goldsmith-Fischman, S.; Kernytsky, A.; Schlessinger, A.;Koh, I.Y.Y.; Alexoc, E.; Honix, B., "Using Multiple Structure Alignments, Fast Model Building, and Energetic Analysis in Fold Recognition and Homology Modeling", *Proteins*. 2003, 53:430-435.

Prasad, J.C.; Comeau, S.R.; Vajda, S.; Camacho, C.J., "Consensus alignment for reliable framework prediction in homolgy modeling", *Bioinformatics*. 2003, 19(13):1682-1691.

Pusey, M.L.; Liu, Z.; Tempel, W.; Praissman, J.; Lin, D.; Wang, B.; Gavira, J.A.; Ng, J.D., "Life in the fast lane for protein crystallization and X-ray cryastallography", *Prog Biophys Mol Biol.* 2005, 88:359-386.

Sali, A. *et al.*, PSA, Unpublished.

Sali, A.; Blundell, T.L., "Comparative Protein Modelling by Satisfaction of Spatial Restraints", *J. Mol. Biol.* 1993. 234:779-815.

Schwede, T.;Kopp, J.; Guex, N.; Peitsch, M.C., "SWISS-MODEL: an automated protein homology-modeling server", *Nucleic Acids Research.* 2003, 31(13):3381-3385.

Shi, J., Harmony3. Unpublished.

Siew, N.; Elofsson, A; Rychlewski, L.;Fischer, D.. "MaxSub: An automated measure for the assessment of protein structure prediction quality", *Bioinformatics*. 2000, 16(9):776-785.

Thompson, J.D.; Higgins, D.G.; Gibson, T.J., "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice", *Nucleic Acids Research.* 1994, 22(22):4673-4680.

Whitelegg, N.R., Rees, A.R., "WAM: an improved algorithm for modeling antibodies on the WEB", *Protein Eng.* 2000, 13(12):819-24.

Wu, C.; Huang, H.; Arminski, L.; Castro-Alvear, J.; Chen, Y.; Hu, Z.; Ledley, R. S.; Lewis, K. C.; Mewes, H.-W.; Orcutt, B.C.; Suzek, B.E.; Tsugita, A.; Vinayaka, C.R.; Yeh, L.-S.; Zhang, J. and Barker, W.C. "The Protein Information Resource: an integrated public resource of functional annotation of proteins", *Nucleic Acids Research.* 2002, 30:35-37.

Wu, T.T.; Kabat, E.A., "An analysis of the sequences of the variable regions of Bence Jones proteins and myeloma light chains and their implications for antibody complementarity", *J Exp Med.* 1970, 132(2):211-250.

Xiang, J., JACKAL. May 2002<http://trantor.bioc.columbia.edu/programs/jackal/>

XLSTAT. Vers. 2006.2. 1 Apr. 2006 <http://xlstat.com>.

Ye, Y.; Godzik, A., *"Flexible structure alignment by chaining aligned fragment pairs allowing twists", Bioinformatics.* 2003, 19(Suppl 2):II246-II255

Zemla, A. "LGA: A method for finding 3D model similarities in protein structures", *Nucleic Acids Research.* 2003, 31(13):3370-4.

Appendix

```
CLUSTAL W (1.83) multiple sequence alignment

A03907          QVQLKESGPGLVAPSQSLSITCTVS----------WVRQPPGKGLEWLG----------------RLSISKDNSKSQVFLKM---HTDD...
AB063756        QITLKESGPTLVKPAQTLTLTCSFS----------WIRQPPGKALEWLA----------------RLAITKDTSKNQVVLRM---DPVD...
AB027447        QVQLQEWGPGLARPSGTLSLTCSVS----------WIRQPPGKGLEWIG----------------RVTINVDTSNNQFSLNL---TAAD...
AB063705        QVQLQESGPGLVKPSDTLSLTCTVS----------WIRQAPGKGLEWIG----------------RVTISVDTAKNQFSLNL---TAAD...
AB063721        QVQLVQSGPGLVKPSETLSLTCTVS----------WIRQPPGKGLEWIG----------------RVTISVDTSKNQFSLKL---TAAD...
AB063791        QVQLQESGPGLVKPSETLSLTCTVS----------WIRQPPGKGLEWIG----------------RVTISVDTSKNQFSLKL---TAAD...
AB063723        QVQLQESGPGLVKPSETLSLTCTVS----------WIRQPPGKGLEWIG----------------RVTISVDTSKKQFSLKL---TAAD...
AB063744        QVQLQESGPGLVKPSETLSLTCTVS----------WIRQPPGKGLEWIG----------------RVTISVDTSRKQFSLKL---TAAD...
AB063729        QVQLQQSGPGLVKPSETLSLTCTVS----------WIRQPPGKGLEWIG----------------RVTISVDTSKKQFSLKL---TAAD...
AB063661        QVQLQQWGAGLLKPSETLSLTCAVY----------WIRQPPGKGLEWIG----------------RVTISVDTSKNQFSLKL---TAAD...
AB063656        QVQLQESGPGLVKPSGTLSLTCAVS----------WVRQPPGKGLEWIG----------------RVTISVDKSKNQFSLKL---TAAD...
AB063787        QVQLQESGPGLVKPSETLSLSCAVS----------WVRQPPGKGLEWIG----------------RVAISIDTSRNQFFLHL---TGAD...
Query           EVKLQESGPSLVKPSQTLSLTCSVTGDSITSDFWSWIRQFPGNRLEYMGFVQYSGETAYNPSLKSRISITRDTSKNQYYLDLNSVTTED...
```

Key: ■ – Framework 1 ■ – Framework 2 ■ – Framework 3

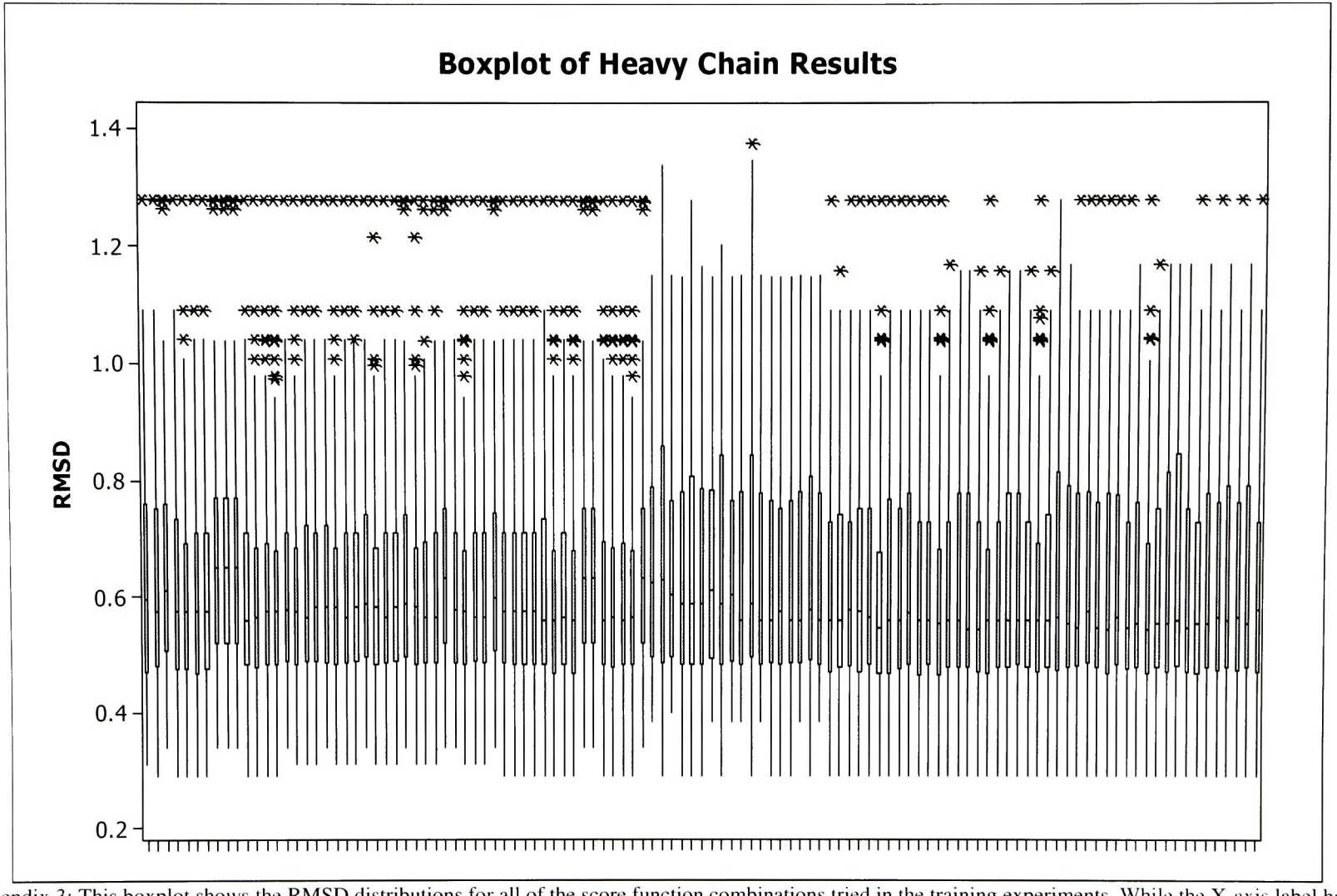
Appendix 1: This is a portion of a one of the profile alignments that are used to Kabat number the antibody sequence. The accession number for each entry is shown at the far left of each line. Following the accession is some whitespace and then the sequence. The first block of characters, shown in red, is framework 1. This region is followed by CDR 1 represented with all '-' characters. Next is framework 2 shown in blue and CDR 2 shown with all '-' characters. The last section is Framework 3 and is shown in green. This region has a known insertion and this insertion is shown with '-' characters. The last sequence is this alignment represents what a query sequence looks like when it is aligned to the profile. As it shows, the CDR residues align to the gaps and the frameworks align to the framework characters in the profile.

```
>a scoring scheme for HV based on proximity to CDRs from the high resolution set with 80% PID or less
3.527345455 3.6571      4.488816667 3.571766667 7.743733333 7.315216667 11.29736667 14.31901667
15.538      15.48546667 16.97916667 13.89301667 14.19645    13.54885    11.01266667 10.27991667
9.624266667 10.5295     11.13728333 9.238716667 8.445033333 4.604566667 6.23315     3.412183333
1.329583333 0           0           0           0           0           0           0
0           0           0           1.328716667 4.089516667 3.919666667 9.072116667 8.502416667
12.57148333 13.81058333 10.10266667 8.80215     6.019316667 3.645633333 2.886016667 2.90685
1.32975     0           0           0           0           0           0           0
0           0           0           0           0           0           0           0
0           1.329766667 3.076616667 3.705333333 2.975683333 3.825583333 3.257583333 4.255433333
3.365833333 6.095883333 5.934883333 3.347033333 3.7499      3.676733333 6.124266667 6.1964
6.871233333 5.93565     7.86205     10.64708333 6.835616667 5.356116667 9.500783333 9.121316667
9.749716667 6.18835     3.87605     3.242883333 2.795666667 1.327483333 0           0
0           0           0           0           0           0           1.327533333 4.442666667
7.663616667 9.464066667 10.63558333 13.57905    11.14641667 13.91441667 11.3323     15.85305
18.59084483
```

```
>Scoring system for heavy chain with 1 for Framework and 0 for CDR residues
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1
1 1 1 1 1
```

Key: ■ – Framework 1 ■ – Framework 2 ■ – Framework 3 ■ – Framework 4

Appendix 2: Examples of two score function files are shown here. Both score function files show all four framework regions, underscored in color, as well as all four CDR regions, represented by 0's. The top score function contains distances and is from score function E. These distances are transposed into scores within runmodeler.pl. The second score function file is from the heavy chain score function B. This file also has all four frameworks underscored in color and CDRs represented by 0's. The difference between these two is that the values from the second file are applied directly as scores to each matching residue at the given position, where as the values from the first file are transposed into scores and then applied. The key to both of these files is that they contain exactly 110 entries, the correct number for a perfectly ungapped Kabat numbered heavy chain antibody.



Appendix 3: This boxplot shows the RMSD distributions for all of the score function combinations tried in the training experiments. While the X-axis label has been removed for readability it should be noted that the first box is score function A and the second box is score function B.